Network Optimization Models II

Kruskal's Algorithm

Shortest Path Algorithm

Applications of Shortest Paths

Class 21

April 5, 2023

Network Optimization Problems

Minimal Spanning Tree

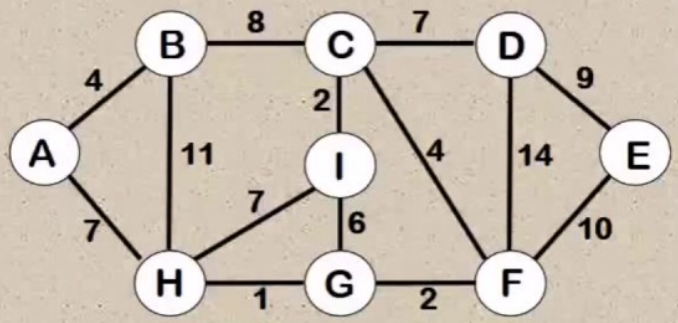
Shortest Path

Maximum Flow

Kruskal's Algorithm
(Minimum Spanning Tree)

Graph:

- Connected
- Weighted
- Undirected



Minimum Spanning Tree:

- Minimum possible total edge weight
- All the vertices connected
- No cycles
- Edges are a sub set of the graph edges

Number of Vertices = 9

Stop when number of edges of the spanning tree = (# of vertices -1) = 9-1 = 8

Kruskal's Algorithm

Kruskal's algorithm is an algorithm in graph theory that finds a minimum spanning tree for a connected weighted graph. Kruskal's algorithm is an example of a greedy algorithm.

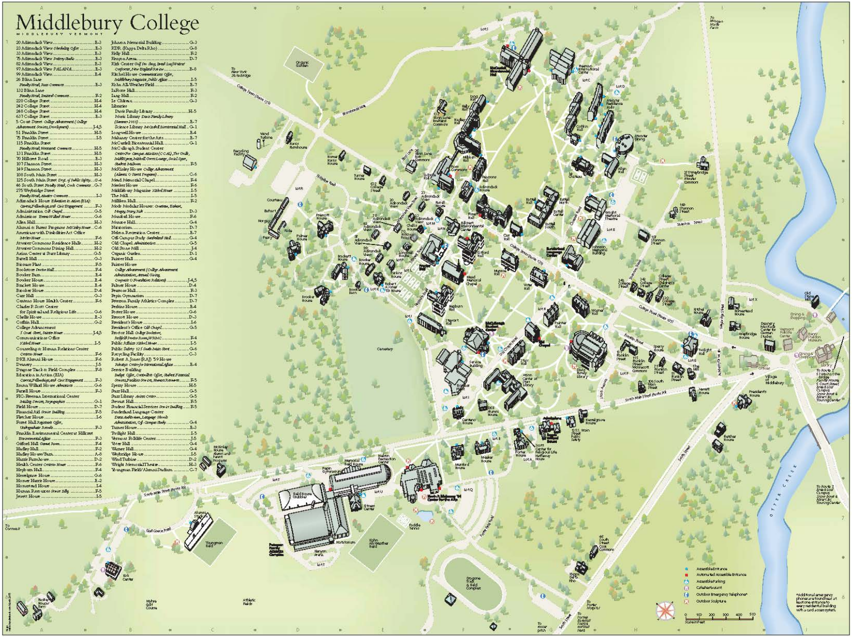
Kruskal's Algorithm in One Sentence: At each stage, add the cheapest edge that connects two nodes not already connected

More formally,

- Create a forest F (a set of trees), where each vertex in the graph is a separate tree
- Create a set S containing all the edges in the graph
- while S is nonempty:
  - Remove an edge with minimum weight from S
  - If that edge connects two different trees, then add it to the forest, combining two trees into a single tree
  - Otherwise discard that edge

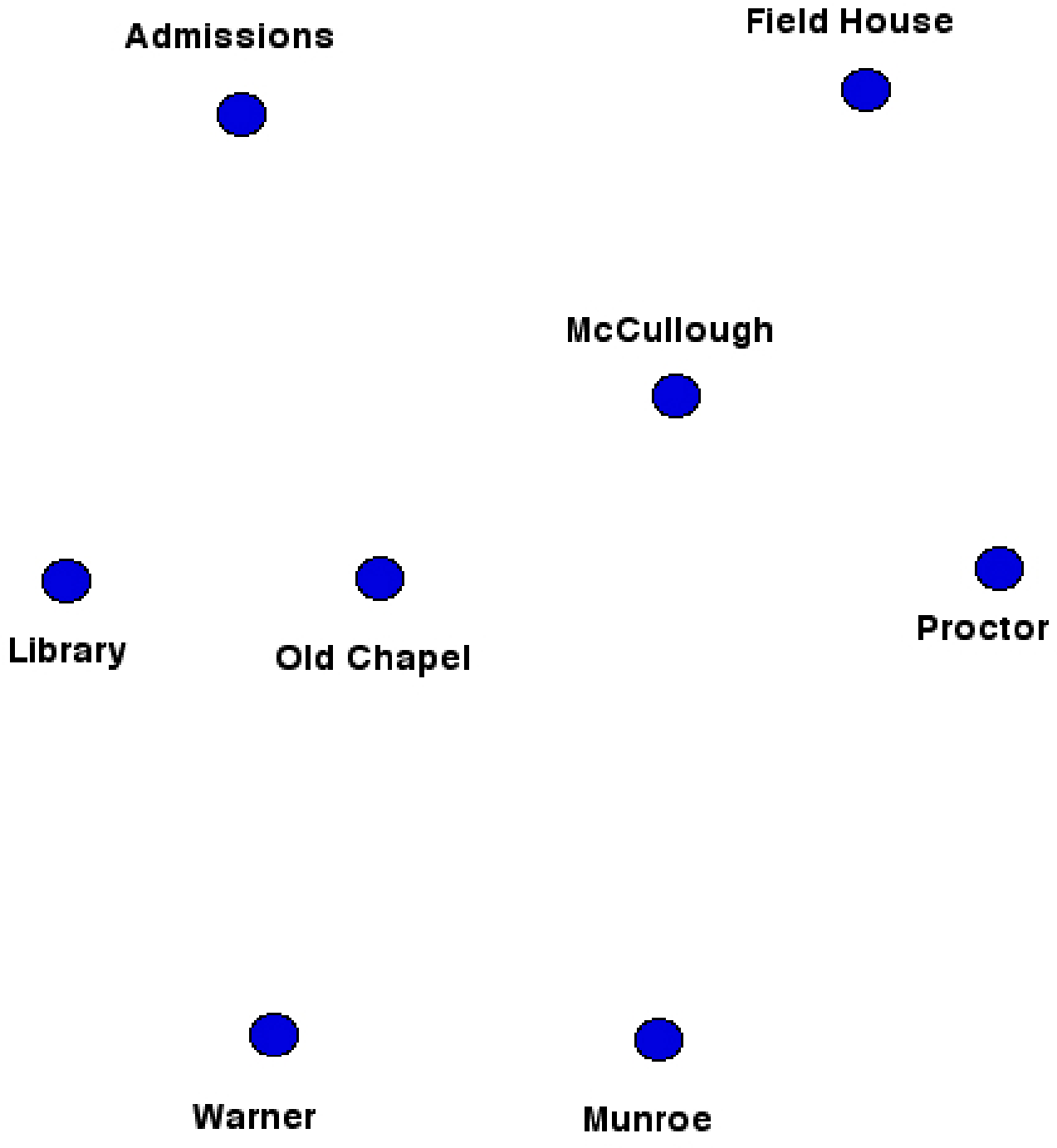
Middlebury College

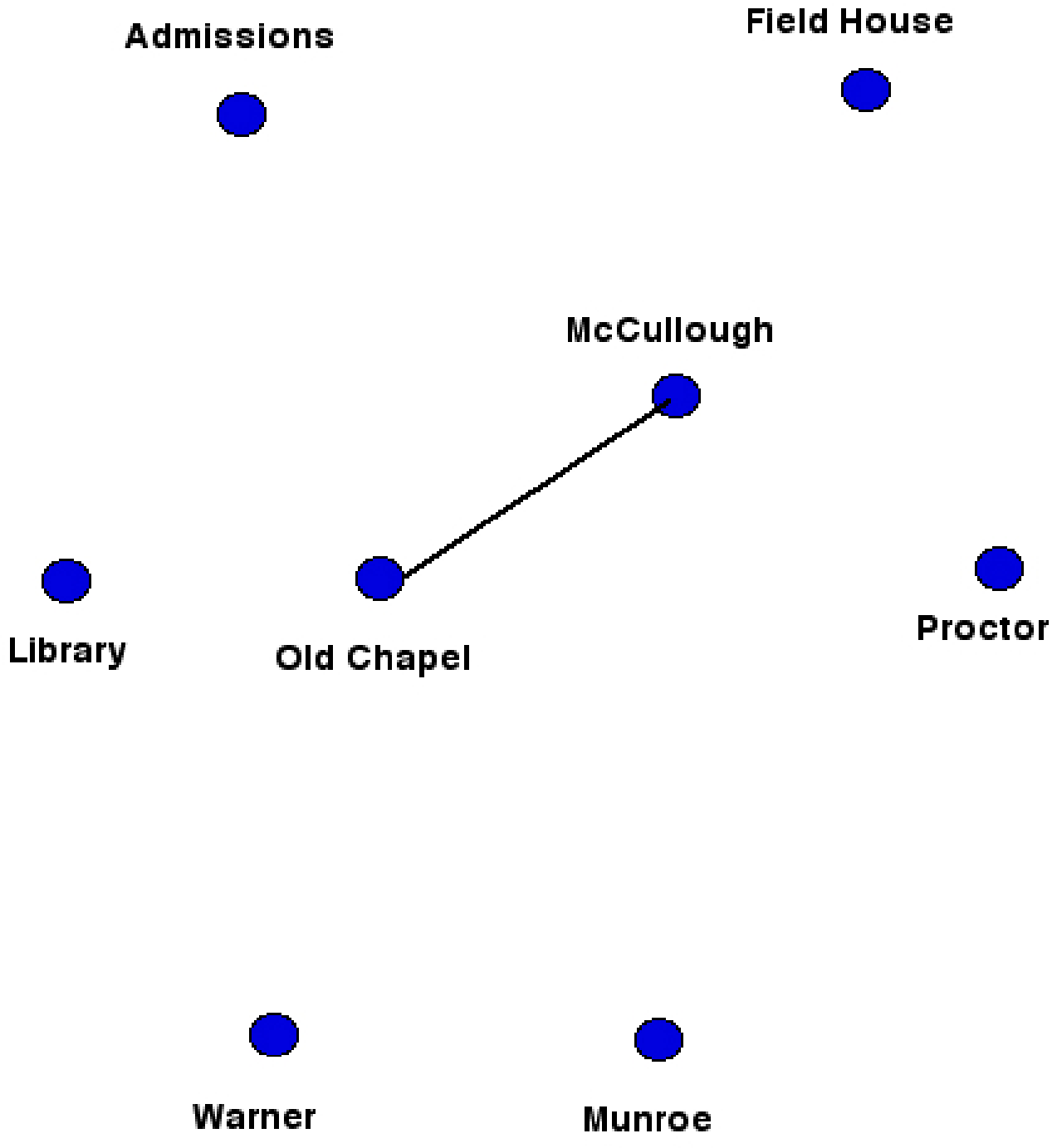
MIDDLEBURY VERMONT

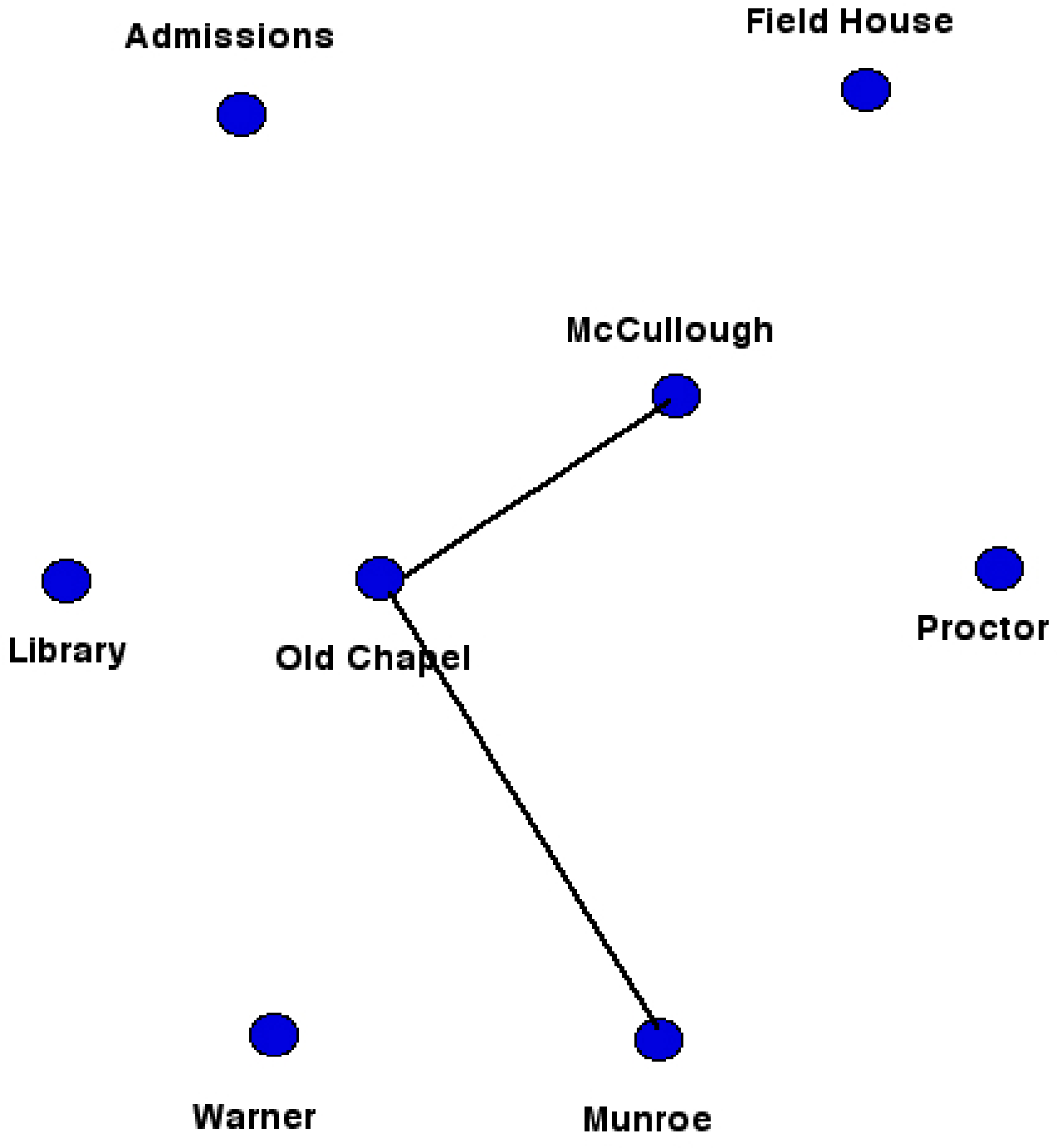


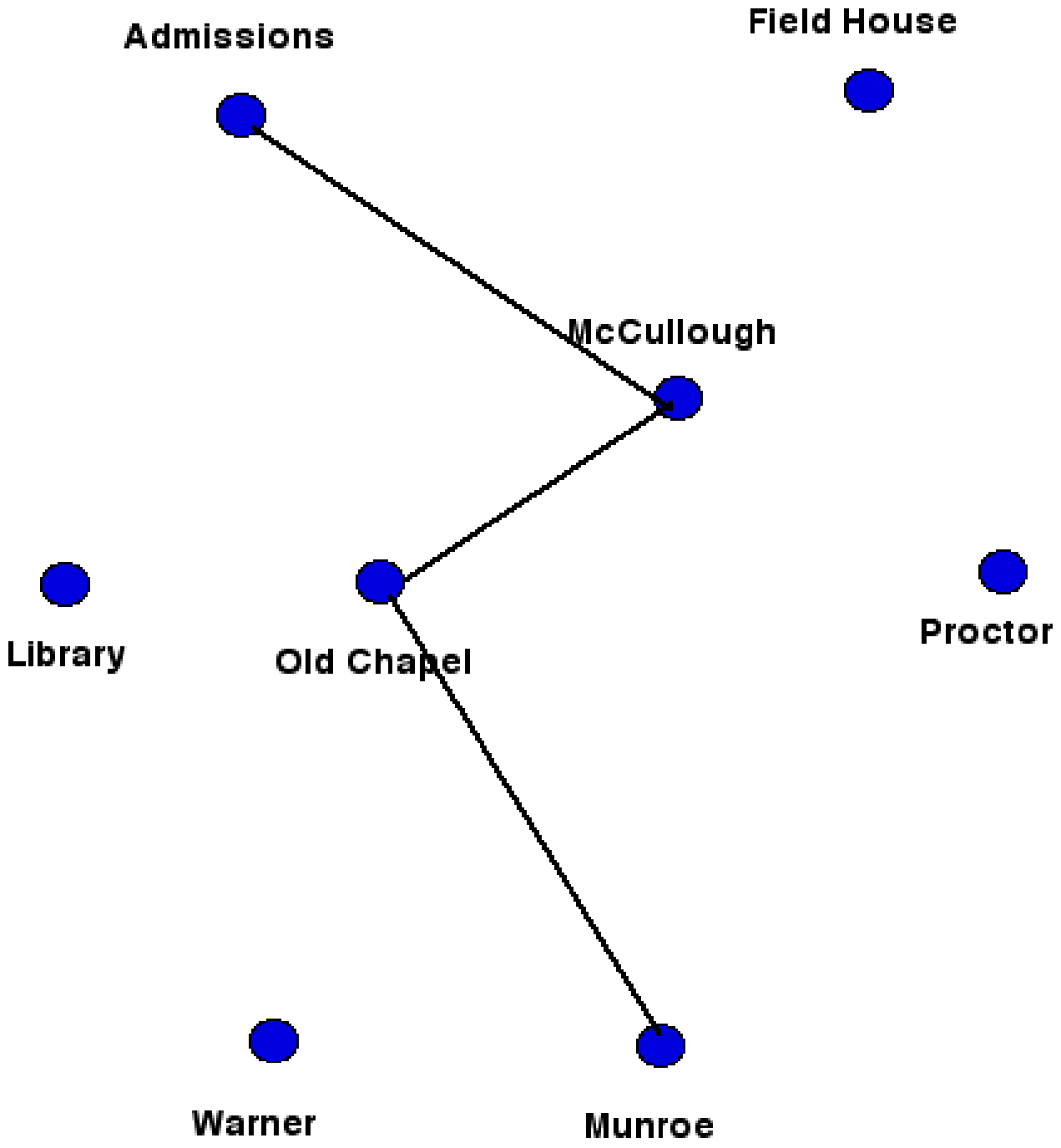
20 Adirondack View E-3
23 Adirondack View *Scheduling Office* E-3
33 Adirondack View E-3
75 Adirondack View *Pottery Studio* E-3
82 Adirondack View E-3
97 Adirondack View *PALANA* E-3
99 Adirondack View E-4
26 Blinn Lane
Faculty Head, Ross Commons E-3
132 Blinn Lane
Faculty Head, Brainerd Commons F-2
220 College Street H-4
242 College Street H-4
248 College Street H-4
637 College Street E-3
5 Court Street *College Advancement (College Advancement Services, Development)* J-4,5
51 Franklin Street H-5
75 Franklin Street I-5
115 Franklin Street
Faculty Head, Wonnacott Commons H-5
131 Franklin Street H-5
70 Hillcrest Road E-3
107 Shannon Street H-3
149 Shannon Street H-3
108 South Main Street H-3
125 South Main Street *Dept. of Public Safety* G-6
46 South Street *Faculty Head, Cook Commons* G-7
275 Weybridge Street
Faculty Head, Atwater Commons I-3
Adirondack House *Education in Action (EIA): Careers, Fellowships, and Civic Engagement* F-3
Administration *Old Chapel* G-5
Admissions *Emma Willard House* G-6
Allen Hall H-3
Alumni & Parent Programs *McKinley House* C-6
Americans with Disabilities Act Office
Meeker House F-6
Atwater Commons Residence Halls H-2
Atwater Commons Dining Hall H-2
Axinn Center at Starr Library G-5
Battell Hall G-3
Biomass Plant F-5
Bookstore *Proctor Hall* F-4
Bowker Barn E-4
Bowker House E-4
Brackett House E-4
Brooker House D-4
Carr Hall G-3
Centeno House Health Center F-6
Charles P. Scott Center
for Spiritual and Religious Life G-6
Chellis House E-3
Coffrin Hall G-2
College Advancement
5 Court Street, Painter House J-4,5
Communications Office
Kitchel House I-5
Counseling & Human Relations Center
Centeno House F-6
DKE Alumni House F-6
Deanery J-5
Dragone Track & Field Complex F-8
Education in Action (EIA)
Careers, Fellowships, and Civic Engagement F-3
Emma Willard House *Admissions* G-6
Farrell House F-3
FIC–Freeman International Center
Mailing Services, Reprographics G-1
Field House D-7
Financial Aid *Service Building* F-5
Fletcher House I-6
Forest Hall *Registrar's Office, Undergraduate Records* F-3
Franklin Environmental Center at Hillcrest
Environmental Affairs F-3
Gifford Hall *Gamut Room* F-4
Hadley Hall F-2
Hadley House/Barn A-8
Harris Farmhouse D-2
Health Center *Centeno House* F-6
Hepburn Hall F-4
Hesselgrave House G-6
Homer Harris House E-2
Homestead House I-4
Human Resources *Service Bldg* F-5
Jewett House I-5
Johnson Memorial Building G-3
KDR (Kappa Delta Rho) G-8
Kelly Hall F-2
Kenyon Arena D-7
Kirk Center *Golf Pro Shop, Bread Loaf/Writers' Conference, New England Review* B-8
Kitchel House *Communications Office, Middlebury Magazine, Public Affairs* I-5
Kohn All-Weather Field E-7
LaForce Hall F-3
Lang Hall F-2
Le Château G-3
Libraries
Davis Family Library H-5
Music Library *Davis Family Library (Summer 2011)* E-7
Science Library *McCardell Bicentennial Hall* G-1
Longwell House E-4
Mahaney Center for the Arts E-7
McCardell Bicentennial Hall G-1
McCullough Student Center
Center For Campus Activities (CCAL), The Grille, MiddXpress, Mitchell Green Lounge, Social Space, Student Mailroom F-5
McKinley House *College Advancement (Alumni & Parent Programs)* C-6
Mead Memorial Chapel F-4
Meeker House F-6
Middlebury Magazine *Kitchel House* I-5
The Mill I-5
Milliken Hall F-2
Mods Modular Houses: *Coutances, Buhart, Norgay, Peary, Ride* D-3
Munford House F-6
Munroe Hall G-4
Natatorium D-7
Nelson Recreation Center E-7
Off-Campus Study *Sunderland Hall* G-4
Old Chapel *Administration* G-5
Old Stone Mill J-4
Organic Garden D-1
Painter Hall G-4
Painter House
College Advancement (College Advancement Administration, Annual Giving, Corporate & Foundation Relations) J-4,5
Palmer House D-4
Pearsons Hall F-3
Pepin Gymnasium D-7
Peterson Family Athletics Complex D-7
Perkins House E-4
Porter House G-6
Prescott House D-3
President's House I-6
President's Office *Old Chapel* G-5
Proctor Hall *College Bookstore, Redfield Proctor Room, WRMC* F-4
Public Affairs *Kitchel House* I-5
Public Safety *125 South Main Street* G-6
Recycling Facility C-3
Robert A. Jones (RAJ) '59 House
Rohatyn Center for International Affairs E-4
Service Building
Budget Office, Controller's Office, Student Financial Services, Facilities Services, Human Resources F-5
Sperry House H-5
Starr Hall G-5
Starr Library *Axinn Center* G-5
Stewart Hall F-5
Student Financial Services *Service Building* F-5
Sunderland Language Center
Dana Auditorium, Language Schools Administration, Off-Campus Study G-4
Turner House E-3
Twilight Hall I-5
Vermont Folklife Center J-5
Voter Hall G-4
Warner Hall G-4
Weybridge House I-5
Wind Turbine D-2
Wright Memorial Theatre H-3
Youngman Field/Alumni Stadium C-7

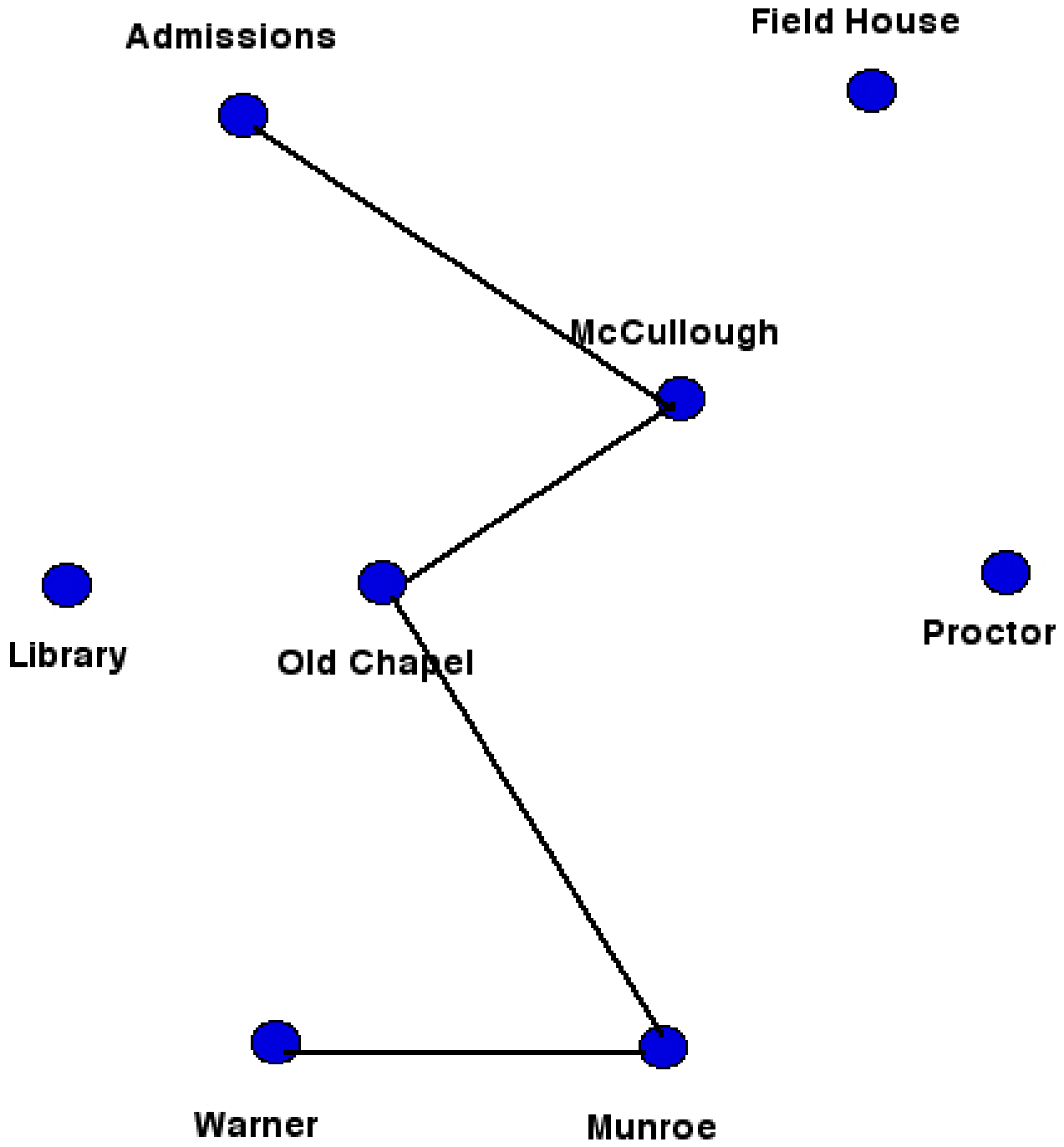
Cost	Arcs
20	Old Chapel-McCullough
21	Old Chapel-Munroe
22	Admissions-McCullough, Warner-Munroe, Field House-Proctor
23	Admissions-Field House, Warner-McCullogh
24	Old Chapel-Field House, McCullough-Proctor
26	Munroe-Proctor
30	Library-Warner
34	Library-Admissions
35	LIbrary-Old Chapel

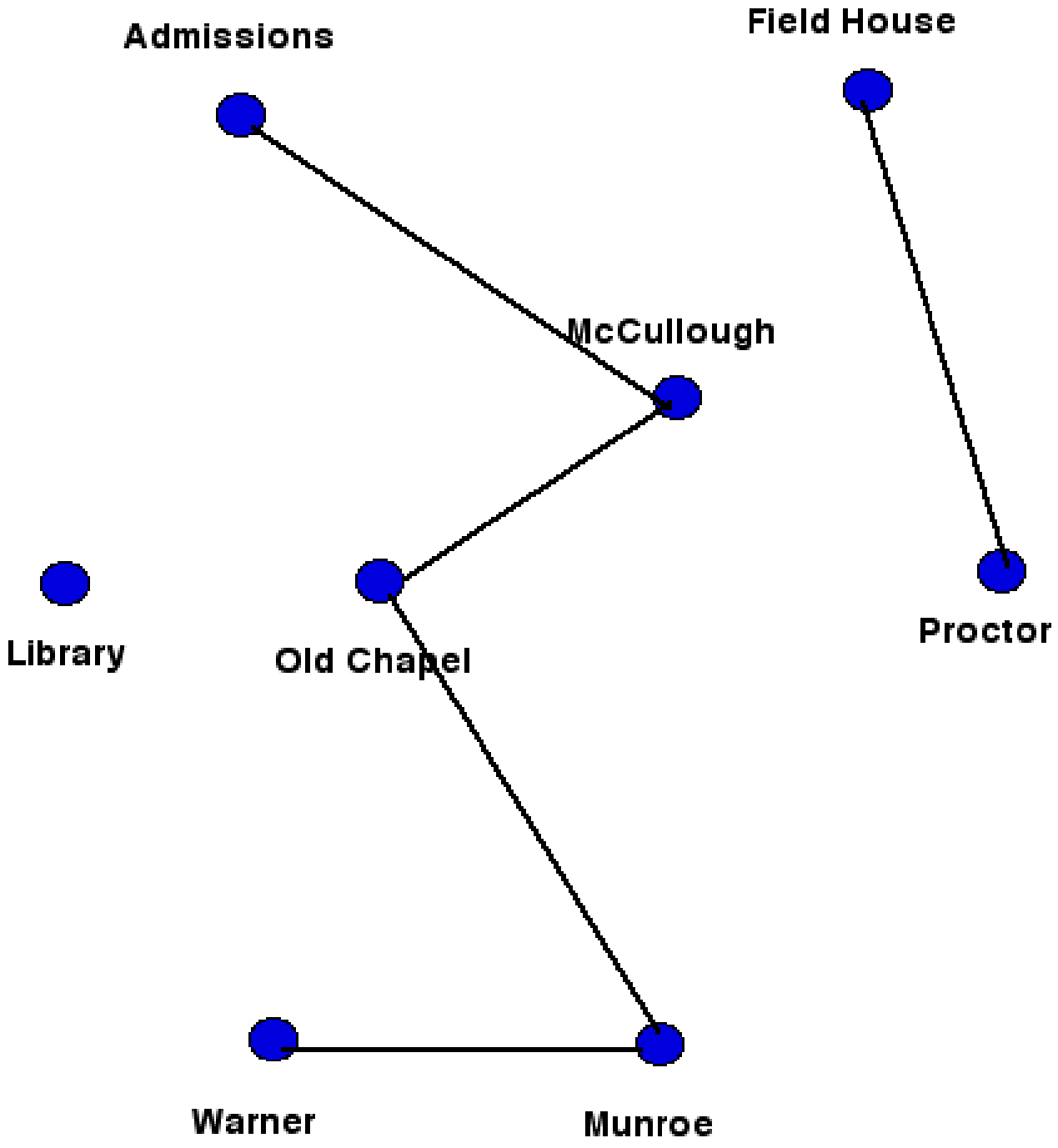

Admissions
Field House
McCullough
Library
Old Chapel
Proctor
Warner
Munroe

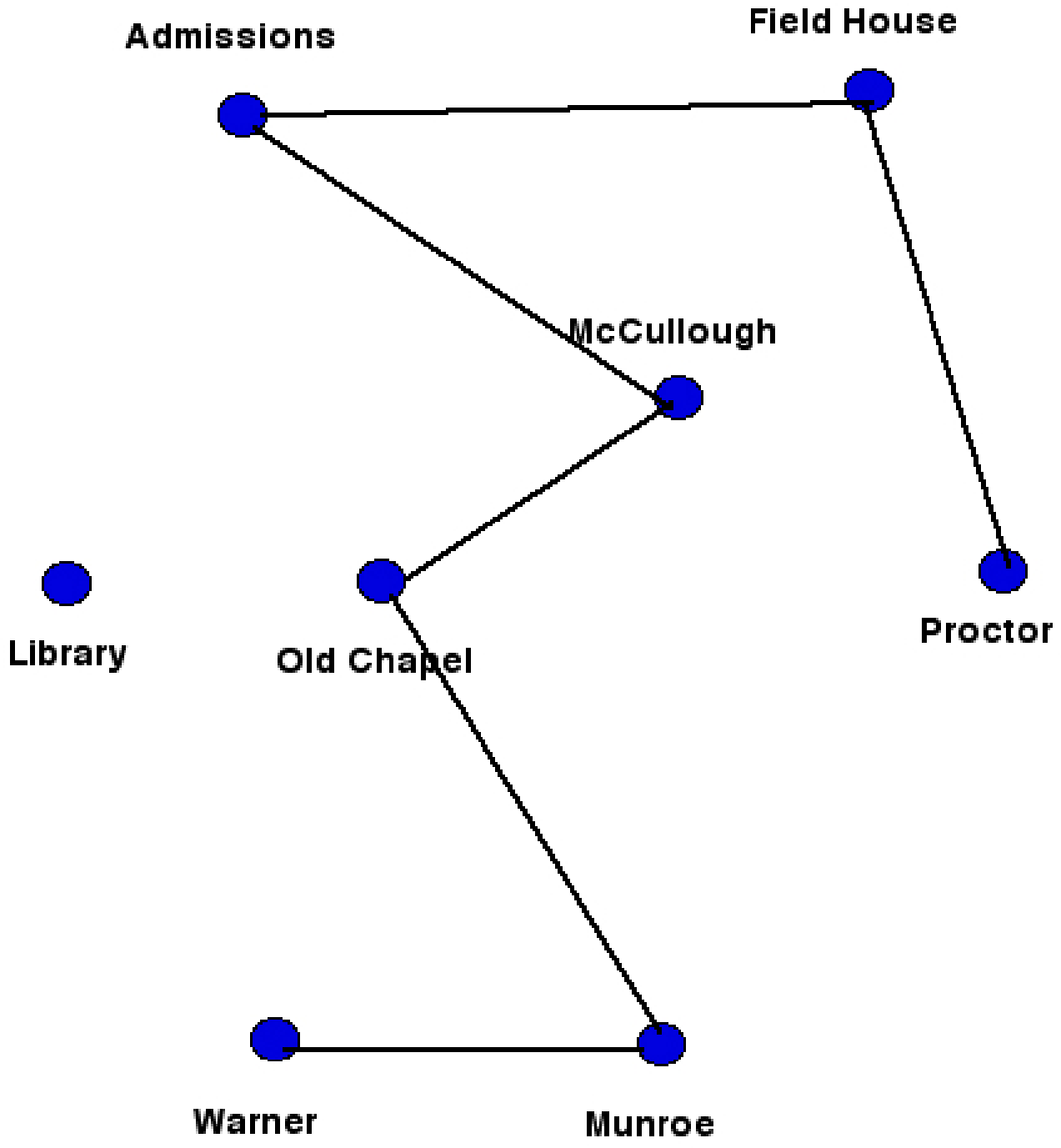

Admissions
Field House
McCullough
Library
Old Chapel
Proctor
Warner
Munroe

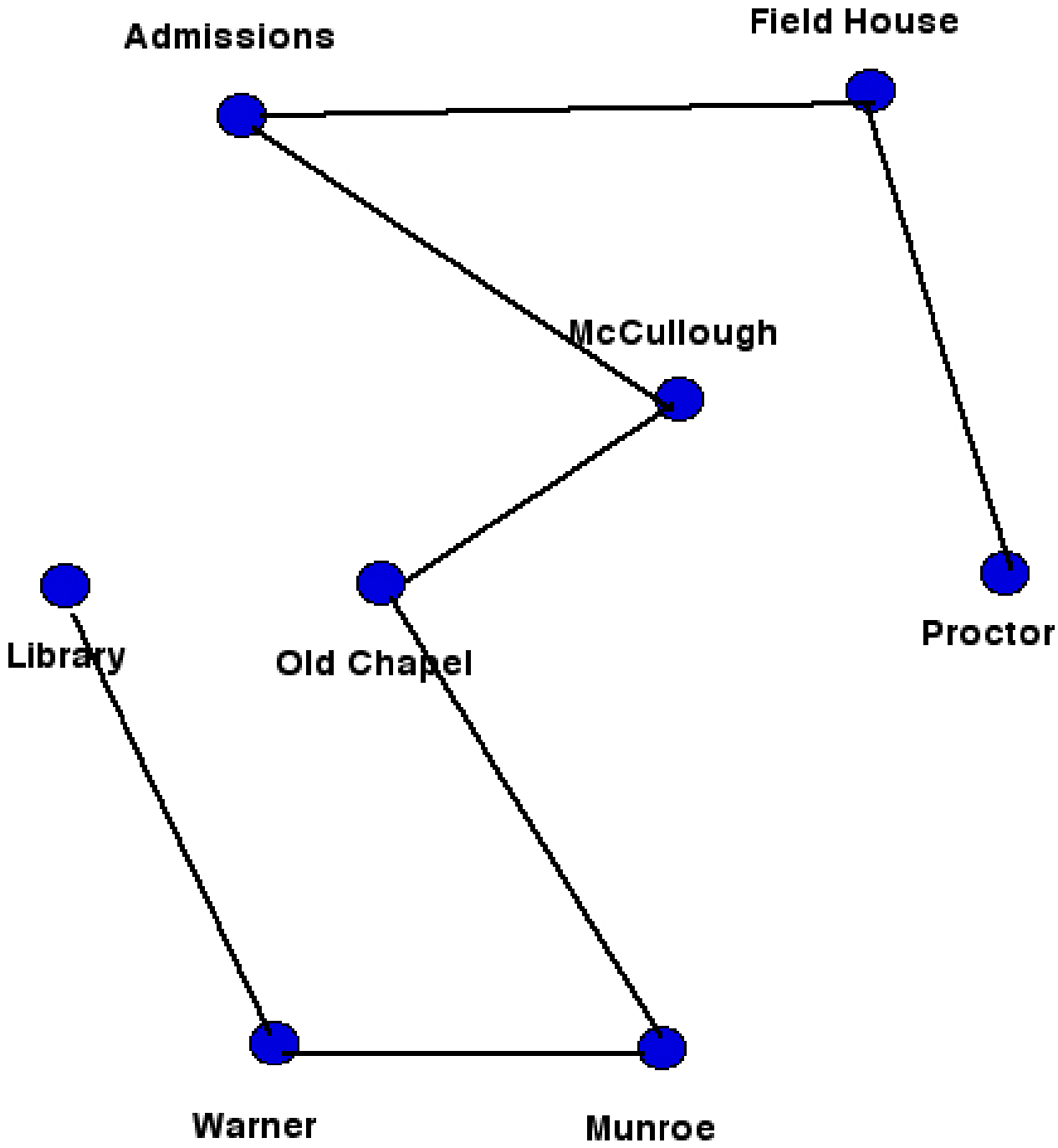

Admissions
Field House
McCullough
Library
Old Chapel
Proctor
Warner
Munroe


Admissions
Field House
McCullough
Library
Old Chapel
Proctor
Warner
Munroe

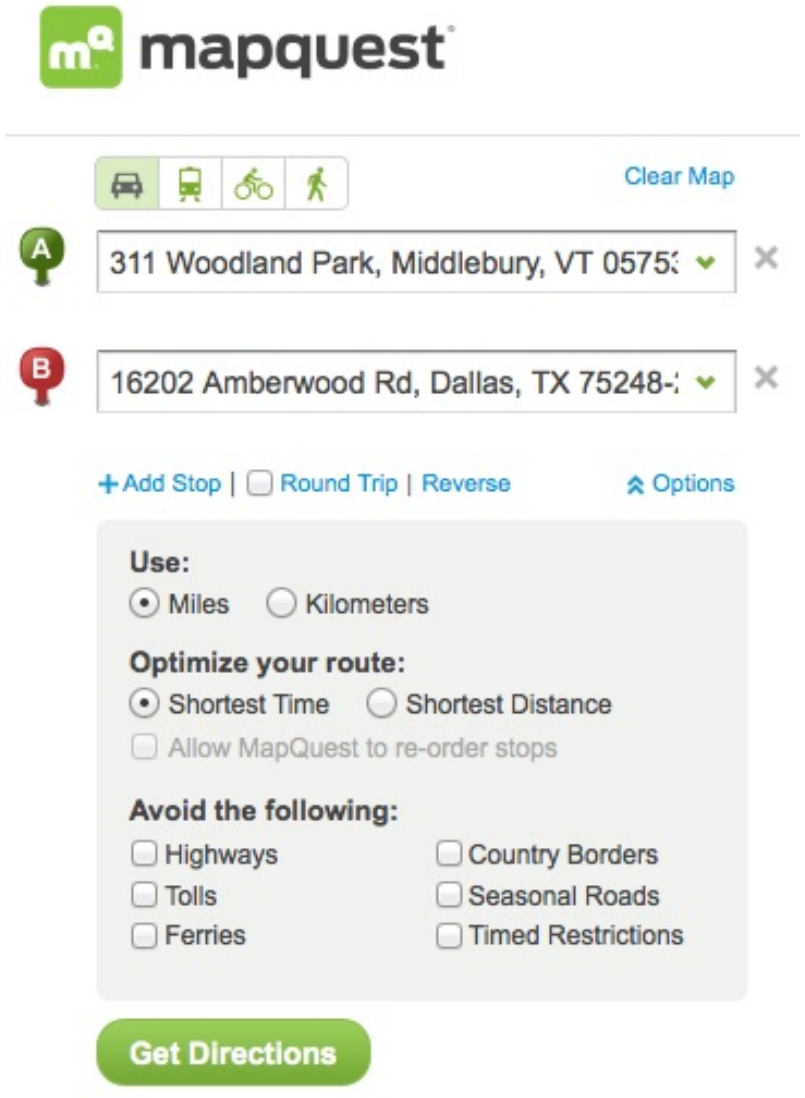

Admissions
Field House
McCullough
Library
Old Chapel
Proctor
Warner
Munroe


Admissions
Field House
McCullough
Library
Old Chapel
Proctor
Warner
Munroe


Admissions
Field House
McCullough
Library
Old Chapel
Proctor
Warner
Munroe


Admissions
Field House
McCullough
Library
Old Chapel
Proctor
Warner
Munroe

Shortest Path Problem



Edsger Wybe Dijkstra

May 11, 1930, Rotterdam, Netherlands

August 6, 2002, Nuenen, Netherlands

A

RIJKSUNIVERSITEIT TE LEIDEN

STUDIEJAAR 1951—1952

No. 1209

De RECTOR MAGNIFICUS verklaart, dat

mej. / de heer E. W. Dijkstra

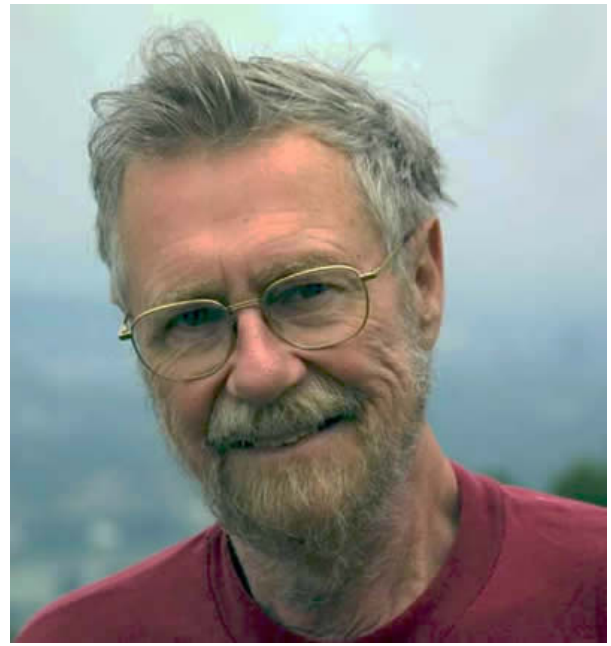
als STUDENT is ingeschreven en mitsdien gedurende bovenvermeld studiejaar toegang heeft tot alle lessen der Rijksuniversiteiten, alsmede tot de daarbij behorende inrichtingen en verzamelingen.

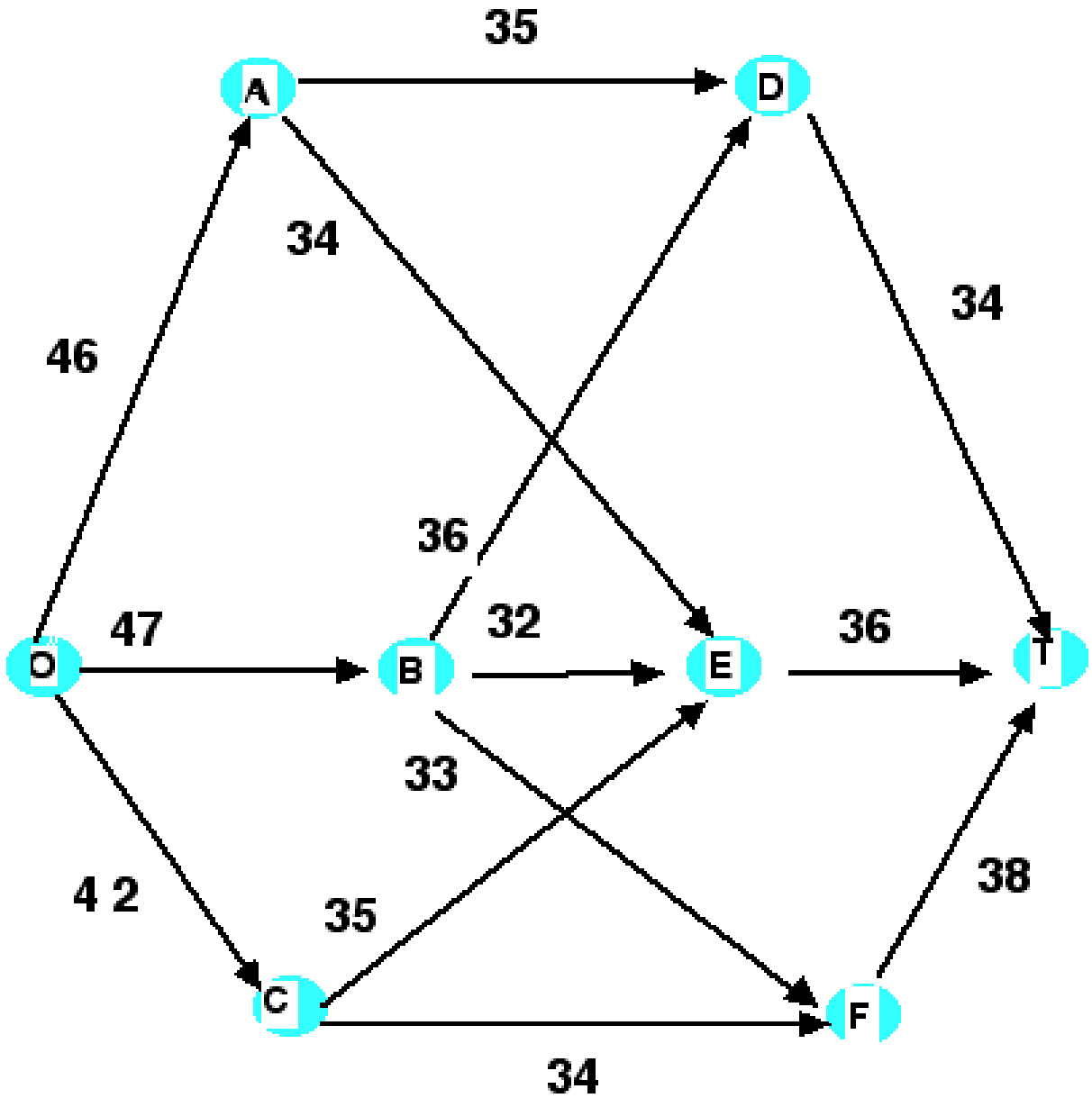
LEIDEN, 15 OCT 1951

[Handtekening van de ingeschrevene (ne varietur)]:

10454 - 51

De ingeschrevene is verplicht deze kaart bij zich te hebben wanneer hij(zij) een college, lurichting of verrandting bezoekt en haar op verzoek van een controlerend ambtenaar te tonen.




35
A
D
34
46
34
36
47
32
36
O
B
E
T
33
4 2
35
38
C
F
34

Dijsktra's Algorithm For The Shortest-Path Problem (1959)

We have a connected network with two special nodes designated the **Origin** and the **Destination** and a non-negative distance assigned to each link

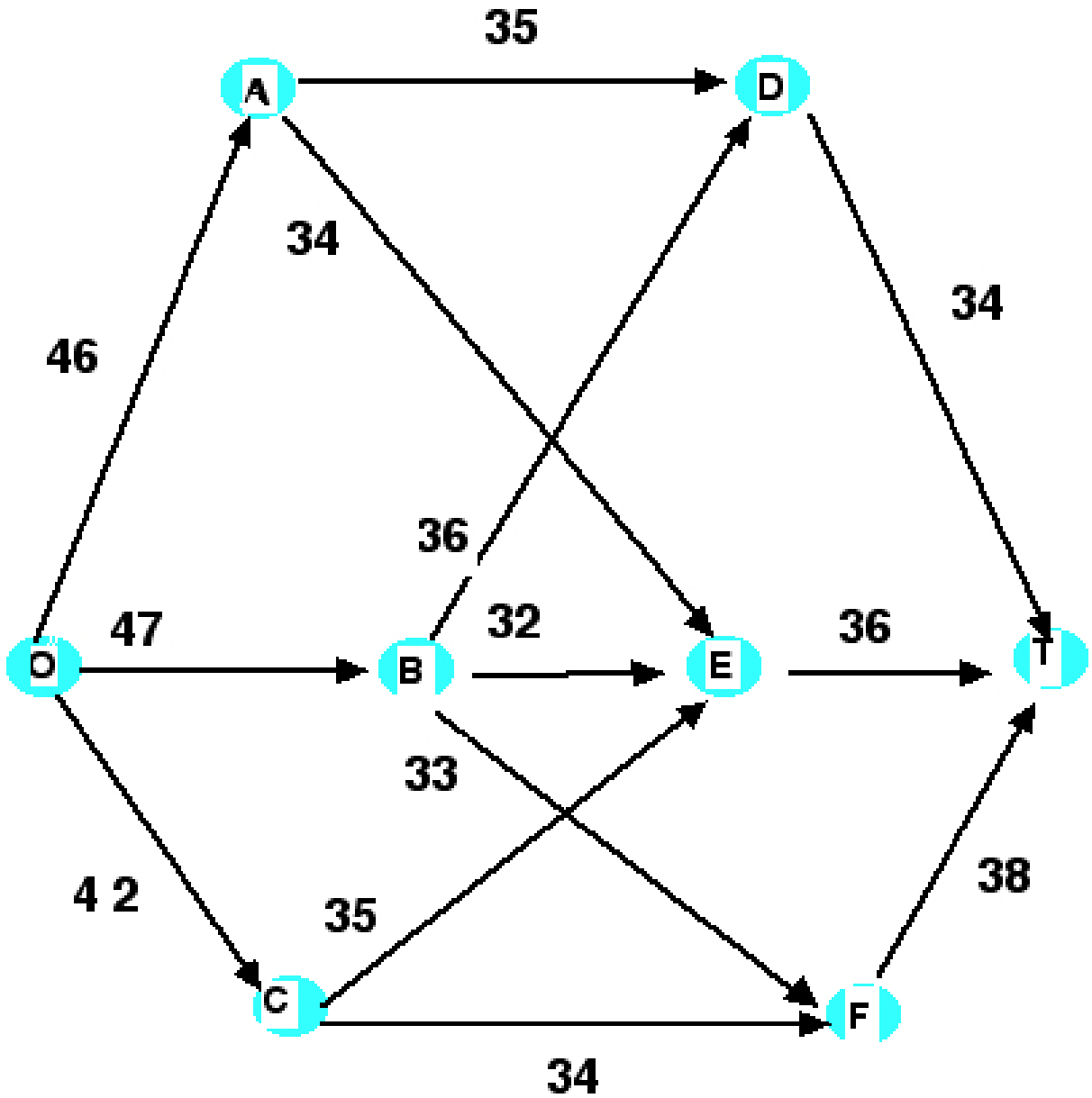
.

Objective of the *n*th iteration: Find the nth nearest node to the **Origin**. Repeat for $n = 1,2,3,..$ until the nth nearest node is the **Destination**.

Input for the *n*th iteration: $n-1$ nearest nodes to the **Origin** solved for at the previous iterations, including their shortest path and distance from the **Origin**. These nodes, together with the **Origin**, will be called **solved nodes**; the others are **unsolved nodes.**

Candidates for the *n*th nearest node: Each solved node that is directly connected by a link to one or more unsolved nodes provides one candidate: the unsolved node with the **shortest** connecting link. (Ties provide additional candidates)

Calculation of the *n*th nearest node: For each solved node and its candidate, add the distance between them and the distance from the **Origin** to this solved node. The candidate with the smallest such total distance is the *n*th nearest node (ties provide additional solved nodes) and its shortest path is the one generating this distance.


35
A
D
34
46
34
36
47
32
36
O
B
E
T
33
4 2
35
38
C
F
34

n	Solved Nodes Directly Linked	Closest Linked Unsolved	Total Distance Involved	nth Nearest Node	Minimum Distance	Last Link
1	O	C	42	C	42	OC
2	O C	A F	46 42+34 = 76	A	46	OA
3	O C A	B F E	47 42+34 = 76 46+34 = 80	B	47	OB
4	A B C	E E F	46+34 = 80 47+32 = 79 42+34 = 76	F	76	CF
5	A B C F	E E E T	46+34 = 80 47+ 32 = 79 42+35 = 77 76+38 = 114	E	77	CE

n	Solved Nodes Directly Linked	Closest Linked Unsolved	Total Distance Involved	nth Nearest Node	Minimum Distance	Last Link
6	A	D	46+35 = 81			
	B	D	47+36 = 83			
	F	T	76+38 = 114	D	81	AD
	E	T	77+36 = 113			
7	D	T	81+34 =115			
	E	T	77+36 = 113	T	113	ET
	F	T	76+38 = 114			

n	nth Nearest Node	Distance	Last Link
1	C	42	OC
2	A	46	OA
3	B	47	OB
4	F	76	CF
5	E	77	CE
6	D	81	AD
7	T	113	ET

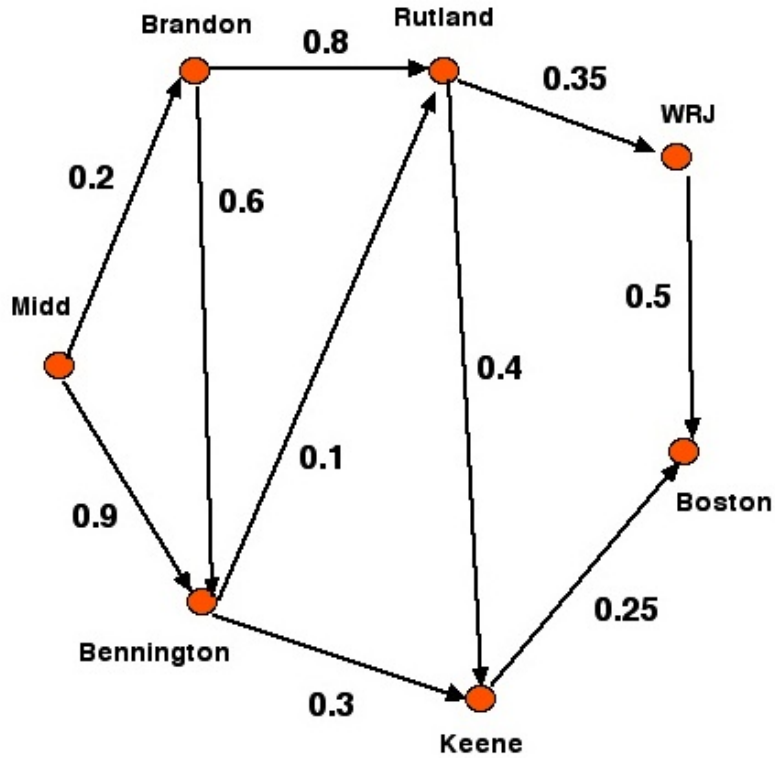
Shortest Path From O to T: OC $\rightarrow$ CE $\rightarrow$ ET

Some Other Shortest Path Problems

Abigail's Most Reliable Route Problem

My daughter Abby often drives between Middlebury and Boston. There are several routes that she can take. Each link is patrolled by different police forces (who do not coordinate with each other). For each link there is an estimated probability of being stopped. The **reliability** of a route is the product of the the reliability of the links in the route. Abby wants to drive very fast, but maximize the probability of not being stopped by the police..

This network displays on each link the probability of not being stopped for speeding.



The **reliability** (that is, probability of not being stopped) along the route Middlebury $\rightarrow$ Brandon $\rightarrow$ Rutland $\rightarrow$ WRJ $\rightarrow$ Boston is $(0.2)(0.8)(0.35)(0.5) = 0.028$ while the reliability of the Middlebury $\rightarrow$ Bennington $\rightarrow$ Keene $\rightarrow$ Boston route is $(0.9)(0.3)(0.25) = 0.0675$

Abby wants to find a route with maximum reliability.
Since the logarithm is an increasing function, if the reliability of Route 1 exceeds the reliability of Route 2, the logarithms have the same relationship:

Let R1 = Reliability of Route 1 and R2 = Reliability of Route 2
Then $R1 > R2$ if and only if $\log(R1) > \log(R2)$.
Furthermore, suppose some route has Reliability R and that route has k links.
Then $R = p_1 \cdot p_2 \cdot p_3 ... \cdot p_k$
and $\log(R) = \log(p_1) + \log(p_2) + ... + \log(p_k)$

Thus Abby's problem, find the route with Maximum R is equivalent to finding the route with Maximum log(R) which, in turn, is equivalent to finding the route with the Maximum sum of the logarithms of the probabilities along its links.

Our problem then is to Maximize a sum

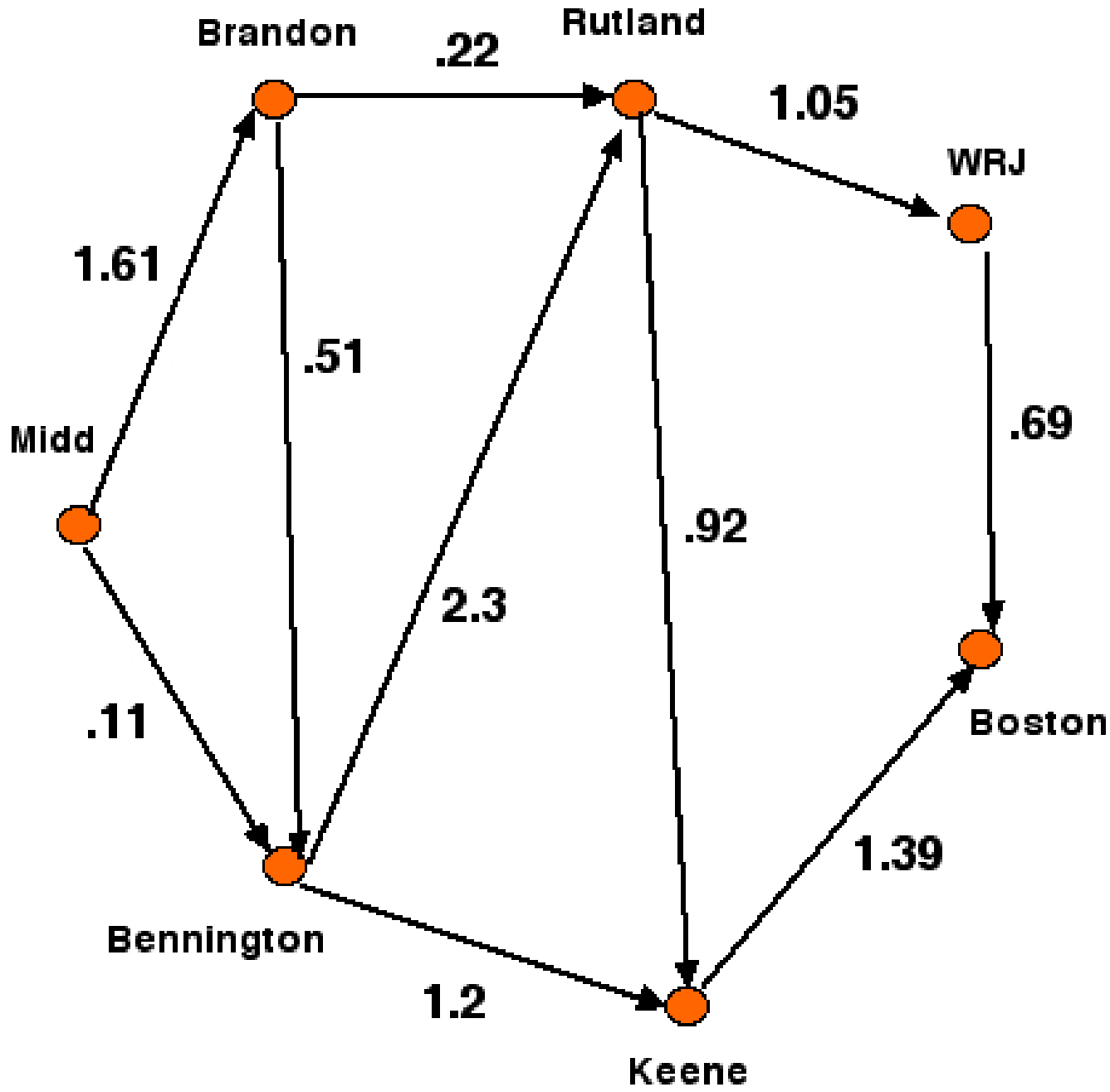
$$\text{Maximize } \log(p_1) + \log(p_2) + ... + \log(p_k)$$
$$\text{which we know is equivalent to}$$
$$\text{MINIMIZE } -(\log(p_1) + \log(p_2) + ... + \log(p_k))$$
$$\text{or MINIMIZE } -\log(p_1) - \log(p_2) + ... - \log(p_k)$$

Now for any link, the probability p_i of not being stopped is a number between 0 and 1. Hence $\log(p_i) < 0$ so $-\log(p_i) > 0$. Thus our original Maximization problem becomes Minimizing the sum of a collection of **POSITIVE** numbers.

An Example

Recall that the reliability of the route
Middlebury $\rightarrow$ Brandon $\rightarrow$ Rutland $\rightarrow$ WRJ $\rightarrow$ Boston is
$R = (0.2)(0.8)(0.35)(0.5) = 0.028$
Then $\log(R) = \log(0.2) + \log(0.8) + \log(0.35) + \log(0.5)$
so $\log R = -1.61 - 0.22 - 1.05 - 0.69$ (rounding to 2 decimal places)
and $-\log R = 1.61 + 0.22 + 1.05 + 0.69$.

Thus to solve Abby's problem, replace each probability along a link with the negative of its logarithm and solve the new shortest path problem.


Brandon
.22
Rutland
1.05
WRJ
1.61
.51
Midd
.69
.92
2.3
.11
Boston
1.39
Bennington
1.2
Keene

An Equipment Replacement Problem

Middlebury College's LIS is developing a replacement policy for its computer servers for a four-year period. At the start of the first year, Middlebury will purchase a server. At the start of each subsequent year, it will decide to keep the server or to replace it. The server will be in service for at least one year but no more than 3 years. This table shows the replacement cost as a function of the period when it is purchased and the years kept in operation:

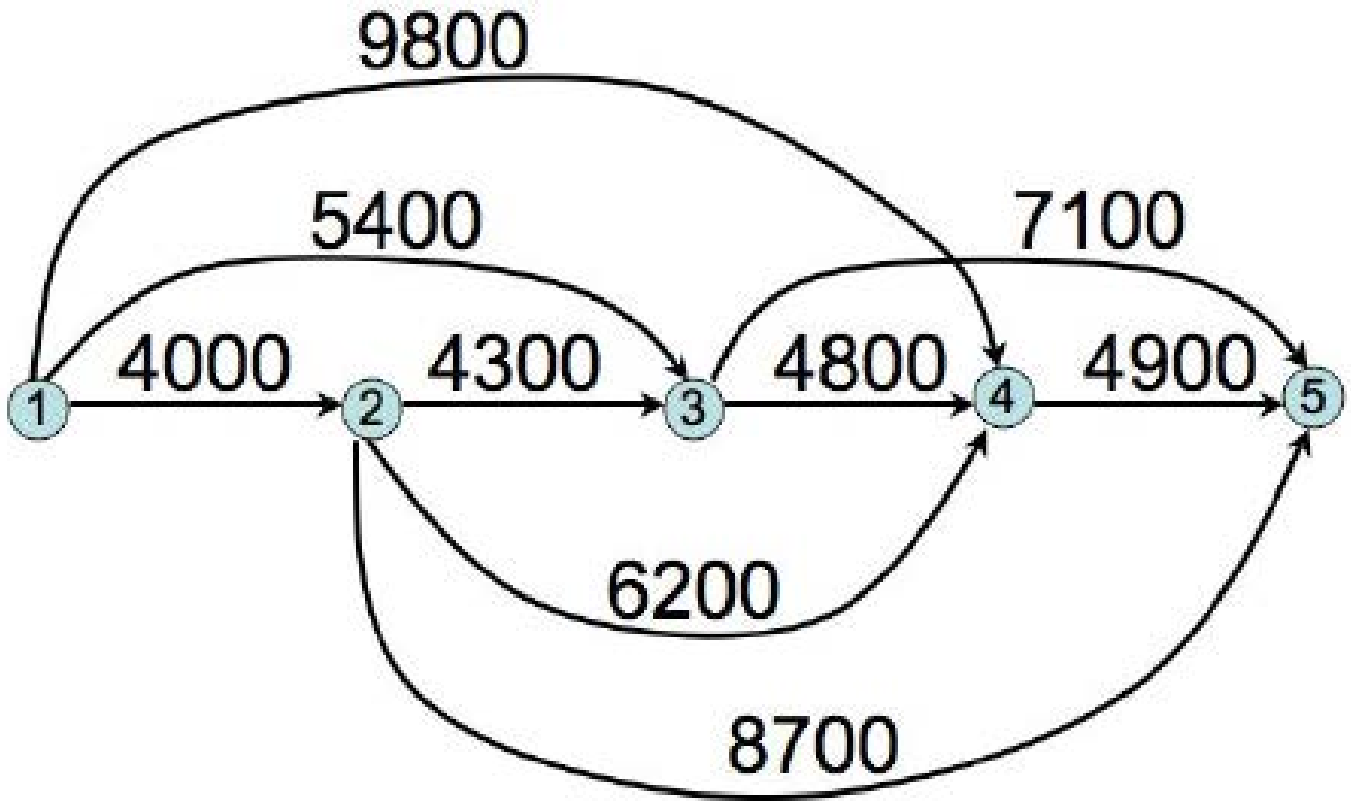
Start of Year	Years in Operation		
	1	2	3
1	4000	5400	9800
2	4300	6200	8700
3	4800	7100	
4	4900		

Problem: Determine the best decision that minimizes the total cost incurred over the 4 year period.

All Possible Decisions

Decision	Cost				Total Cost
By a server in years 1,2,3,4	4000	4300	4800	4900	18,000
By a server in years 1,2,3	4000	4300	7100	x	15,400
By a server in years 1,2,4	4000	6200	x	4900	15,100
By a server in years 1,3,4	5400	x	4800	4900	15,100
By a server in years 1,2	4000	8700	x	x	12,700
By a server in years 1,3	5400	x	7100	x	12,500
By a server in years 1,4	9800	x	x	4900	14,700

Let's Formulate Problem as Shortest Path Problem



The best decision corresponds to the shortest path from node 1 to node 5, which is $1 \rightarrow 3 \rightarrow 5$ with the cost of 5400 + 7100 - 12,500: Purchase a new server in years 1 and 3.

Minimal Spanning Tree Problem

There are n points in the plane whose mutual distances are given.

The problem is to join them with a net in such a way that:

- Any two points are joined to each other either directly or by means of some other points and
- the total length of the net will be minimal.

Proof That Kruskal's Algorithm Works

Let P be a connected, weighted graph and let Y be the subgraph of P produced by the algorithm.

Y cannot have a cycle, since the last edge added to that cycle would have been within one subtree and not between two different trees.

Y cannot be disconnected, since the first encountered edge that joins two components of Y would have been added by the algorithm.

Thus, Y is a spanning tree of P.

Let Y_1 be a minimum spanning tree for P which has the greatest number of edges in common with Y.
If Y_1=Y then Y is a minimum spanning tree.

Otherwise, let **e** be the first edge considered by the algorithm that is in Y but not in Y_1.

Let C_1 and C_2 be the components of F which **e** lies between at the stage when **e** is considered.

Since Y_1 is a tree, Y_1+**e** has a cycle and there is some different edge **f** of that cycle that also lies between C_1 and C_2.

Then Y_2=Y_1 + **e** - **f** is also a spanning tree.

Since **e** is considered by the algorithm before **f**, the weight of **e** is at most equal to the weight of **f**, and since Y_1 is a minimum spanning tree the weights of these two edges must in fact be equal.

Therefore, Y_2 is a minimum spanning tree having more edges in common with Y than Y_1 does, contradicting our assumption about Y_1. This proves that Y must be a minimum spanning tree.