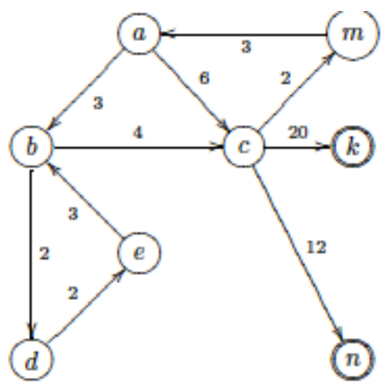
Practice Midterm Problems with Solutions for CPSC 322, Winter 2010 Term 1

The midterm will contain a combination of short questions (see review file for samples of those), and longer problems. The two problems below will help you practice for this second type of questions.

1. Search [22 points]

Consider the following directed graph where *a* is the start node and *k* and *n* are both goal nodes, the true cost function is given by the edge labels, and h is an admissible heuristic function.



node	*h(node)*
a	16
b	16
c	11
d	20
e	18
k	0
m	2
n	0

In answering these questions show your work and justify your conclusions.

1. [**7 points**] A*
   Assume that ties are broken alphabetically.

   (a) What sequence of paths is expanded by A*? (If you like, you can describe each path just by giving the last node in the path.)
   (b) What path is returned?
   (c) What is the cost of this path?
   Answer: A* expands *acmn* and returns the path *acn* which costs 18.

2. [**7 points**] (Depth First) Branch-and-bound search with cycle checking, in which neighbors are expanded according to *f*
   Assume that ties are broken alphabetically.

   (a) What sequence of paths are expanded (or considered for expansion) by branch-and-bound? (If you like, you can describe each path just by giving the last node in the path.)
   (b) What path is returned?
   (c) What is the cost of this path?
   Answer: Branch-and-bound expands *acmnkb* and returns the path *acn* which costs 18.

3. [**8 points**] Consider a complete binary search tree with finite depth *d*, and with only one goal, which is at depth *k* < *d*. The search begins at the root node (let's call this node's depth 1).

   (a) In the best case:
   i. Exactly how many nodes would be expanded by breadth-first search? (including the

goal node)
Answer: 2^{k-1} nodes.

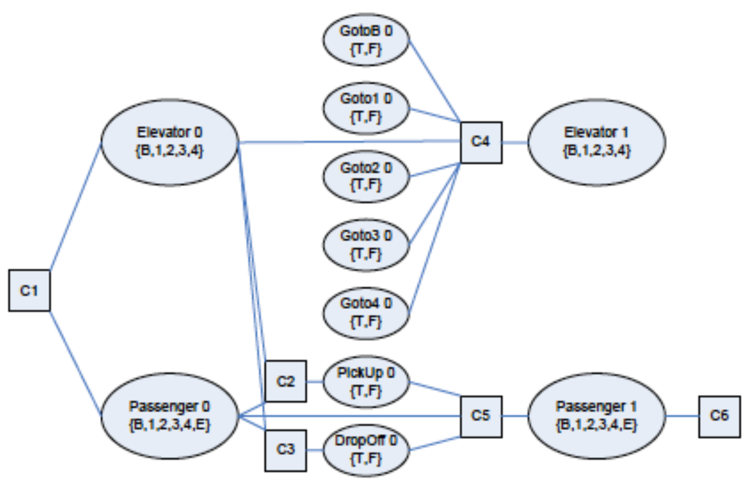
ii. Exactly how many paths would be on the frontier just before the goal node is expanded?
Answer: 2^{k-1} paths.

(b) In the best case:
i. Exactly how many nodes would be expanded by depth-first search? (including the goal node)
Answer: *k* nodes.

ii. Exactly how many paths would be on the frontier just before the goal node is expanded?
Answer: k paths.

2. [22 points] CSP Planning

Consider the one-stage unrolling of a CSP planning problem represented below, in which we have an elevator that can be on different floors and a passenger who can be waiting at different floors or can be in the elevator. Notice that for the sake of simplicity the model is not realistic.



The constraint tables[1] are as follows (with * representing a wildcard or don't-care condition).

C1

Elevator 0	Passenger 0
2	3

[1] Recall that constraint tables list the joint variable assignments that are *allowed.*

C2

Elevator 0	Passenger 0	PickUp 0
3 *	3 *	T F

C3

Passenger 0	Elevator	PickUp 0
E *	4 *	T F

C4

Elevator 0	GotoB 0	Goto1 0	Goto2 0	Goto3 0	Goto4 0	Elevator 1
B	F	F	F	F	F	B
1	F	F	F	F	F	1
2	F	F	F	F	F	2
3	F	F	F	F	F	3
4	F	F	F	F	F	4
*	T	F	F	F	F	B
*	F	T	F	F	F	1
*	F	F	T	F	F	2
*	F	F	F	T	F	3
*	F	F	F	F	T	4

C5

Passenger 0	PickUp 0	DropOff 0	Passenger 1
B	F	F	B
1	F	F	1
2	F	F	2
3	F	F	3
4	F	F	4
E	F	F	E
3	T	F	E
E	F	T	4

C6

Passenger 1
4

1. [**7 points**] Perform arc-consistency on this CSP. (Just cross out the values on the figure above). What is the outcome? What does it mean?
   Answer: Arc consistency will eliminate every value from every domain. By C6, *dom*(Passenger 1) = {4}. By C1, *dom*(Passenger 0) = {3}. Because of this, there is no satisfying assignment for C5, meaning we can eliminate every value from Passenger 0, Passenger 1, PickUp 0 and DropOff 0. Any constraint connected to a variable with an empty domain cannot be satisfied, which means we can eliminate the domain of every connected

variable. No one step solution exists for this planning problem.

2. [**9 points**] Give a STRIPS representation of the problem, including variables, domains, start state, goal states and action descriptions.
   Answer: Variables: Elevator, Passenger
   dom(Elevator) = {B,1,2,3,4}
   dom(Passenger) = {B,1,2,3,4,E}
   Start State: Elevator=2, Passenger=3
   Goal State: Elevator=*, Passenger=4

Action	**Precondition**	**Effect**
GotoB		Elevator=B
Goto1		Elevator=1
Goto2		Elevator=2
Goto3		Elevator=3
Goto4		Elevator=4
PickUp	Elevator=3, Passenger=3	Passenger=E
DropOff	Elevator=4, Passenger=E	Passenger=4

3. [**6 points**] What is the smallest horizon for which this CSP planner can be solved, if a constraint is added to make all actions mutually exclusive? Report the plan that it would find.
   Answer: 4 is the smallest horizon for which this set of CSPs has a solution (with full mutex action constraints). The plan it would find is <<Goto3>,<PickUp>,<Goto4><DropOff>>.