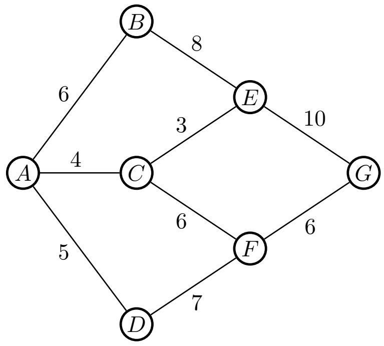
CS4725/CS6705 - Fall 2017
Assignment # 2

SAMPLE SOLUTIONS

1. **(6 marks)**



Consider the state graph shown above. G is the goal state. Each edge is labelled by its cost. An important point for this question is that these edges can be traversed in both directions; for example, you can move from A to C, and you can also move from C to A.

You have been given the heuristic function h below, but you were unable to read the value of $h(C)$. You can assume that it is a non-negative value.

Node n	A	B	C	D	E	F	G
$h(n)$	13	15	?	11	8	6	0

(a) What range of values for $h(C)$ would make h an admissible heuristic? Explain your answer.

In order for h to be admissible, the value of $h(C)$ must not overestimate the true cost of the optimal path from C to G, which is 12 (for the path $C \to F \to G$). There is no minimum value, other than the comment above that you can assume that it is non-negative. Therefore, for h to be admissible, it must be the case that $0 \leq h(C) \leq 12$.

(b) What range of values for $h(C)$ would make h a consistent heuristic? Think carefully about what the definition of *consistent* says. Explain your answer.

From the definition of consistent, we can obtain a number of inequalities that must hold. (Note that C can play the role of n or the role of n' in the definition.)

$$h(C) \leq 4 + h(A) \;\;\to\;\; h(C) \leq 4 + 13 \;\;\to\;\; h(C) \leq 17$$

$$h(C) \leq 3 + h(E) \;\;\to\;\; h(C) \leq 3 + 8 \;\;\to\;\; h(C) \leq 11$$

$$h(C) \leq 6 + h(F) \;\;\to\;\; h(C) \leq 6 + 6 \;\;\to\;\; h(C) \leq 12$$

$$h(A) \leq 4 + h(C) \;\;\to\;\; 13 \leq 4 + h(C) \;\;\to\;\; h(C) \geq 9$$

$$h(E) \leq 3 + h(C) \;\;\to\;\; 8 \leq 3 + h(C) \;\;\to\;\; h(C) \geq 5$$

$$h(F) \leq 6 + h(C) \;\;\to\;\; 6 \leq 6 + h(C) \;\;\to\;\; h(C) \geq 0$$

In order for **all** of these inequalities to hold, it must be the case that $9 \leq h(C) \leq 11$.

2. **(5 marks)**

(a) (3 marks) Suppose that you are solving some search problem in which all action costs are positive. You have two heuristic functions, h_1 and h_2, and you know that both of them are admissible. Assume that $h_1(n) \geq 0$ and $h_2(n) \geq 0$ for all states n.

Based only on the information above, which of the following heuristic functions would **also** be guaranteed to be admissible? Circle **all** that apply.

i. $h_3(n) = 0$ for all states n
ii. $h_4(n) = h_1(n) + h_2(n)$ for all states n
iii. $h_5(n) = 0.5h_1(n) + 0.5h_2(n)$ for all states n
iv. $h_6(n) = 2h_1(n)$ for all states n
v. $h_7(n) = \min\{h_1(n), h_2(n)\}$ for all states n
vi. $h_8(n) = \max\{h_1(n), h_2(n)\}$ for all states n
vii. $h_9(n) = \max\{h_1(n) - h_2(n), h_2(n) - h_1(n)\}$ for all states n

Heuristics h_3, h_5, h_7, h_8 and h_9 would all be guaranteed to be admissible.

(b) (2 marks) Consider the heuristics you selected as admissible in part (b). Based on our discussion in class, which one would lead to the **most efficient** performance when used as the heuristic in A* search? Explain your answer.

The answer is $h_8(n) = \max\{h_1(n), h_2(n)\}$

Because h_1 and h_2 are both admissible, we know that h_8 will **not overestimate** the true cost.

However, h_8 **dominates** the other admissible heuristics in the list.

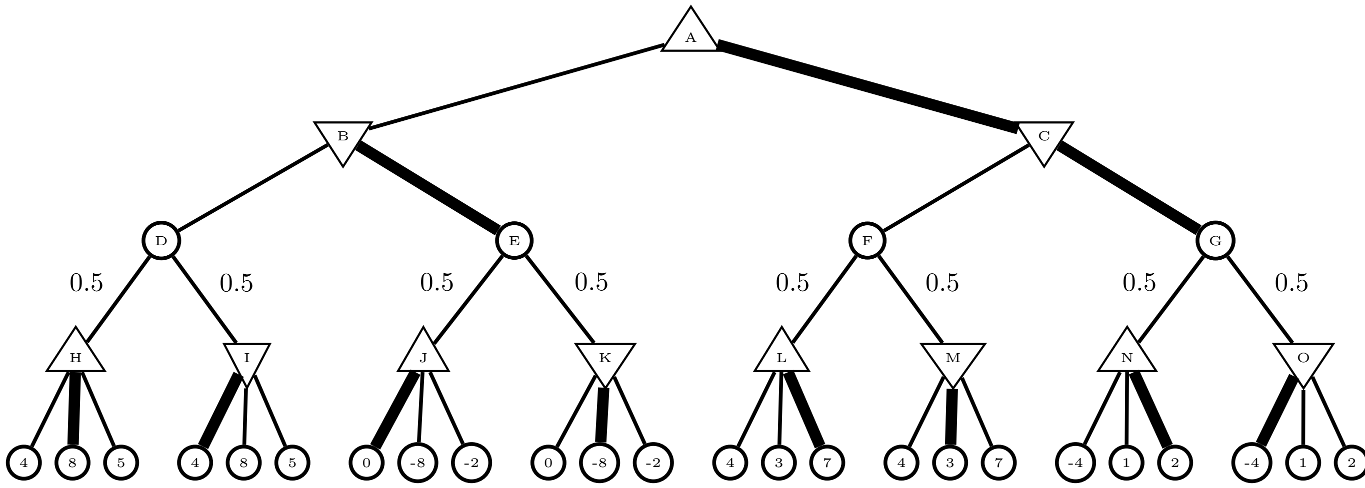
For all n, $h_8(n) \geq h_3(n)$
$h_8(n) \geq h_5(n)$
$h_8(n) \geq h_7(n)$
$h_8(n) \geq h_9(n)$

Therefore, it is the **closest** to the true cost without going over, which translates directly into efficiency.

3. **(8 marks)** Suppose that your friend Minnie has offered to play a game with you. You get to play the first move, after which Minnie gets to play a move. After that, you flip a coin. A coin flip of 'heads' means that you get to make the final move of the game; a coin flip of 'tails' means that it's Minnie's move.

   The game tree is shown below, with your utility values shown in the bottom row for the different possible game outcomes. You can think of these as the number of dollars you would win or lose as a result of each outcome.

   Use the expectiminimax technique to solve this game tree.



- What is the expectiminimax value of each node?

$$
\begin{aligned}
&value(H) = \max\{4,8,5\} = 8\\
&value(I) = \min\{4,8,5\} = 4\\
&value(J) = \max\{0,-8,-2\} = 0\\
&value(K) = \min\{0,-8,-2\} = -8\\
&value(L) = \max\{4,3,7\} = 7\\
&value(M) = \min\{4,3,7\} = 3\\
&value(N) = \max\{-4,1,2\} = 2\\
&value(O) = \min\{-4,1,2\} = -4\\
&value(D) = 0.5(8) + 0.5(4) = 6\\
&value(E) = 0.5(0) + 0.5(-8) = -4\\
&value(F) = 0.5(7) + 0.5(3) = 5\\
&value(G) = 0.5(2) + 0.5(-4) = -1\\
&value(B) = \min\{6,-4\} = -4\\
&value(C) = \min\{5,-1\} = -1\\
&value(A) = \max\{-4,-1\} = -1
\end{aligned}
$$

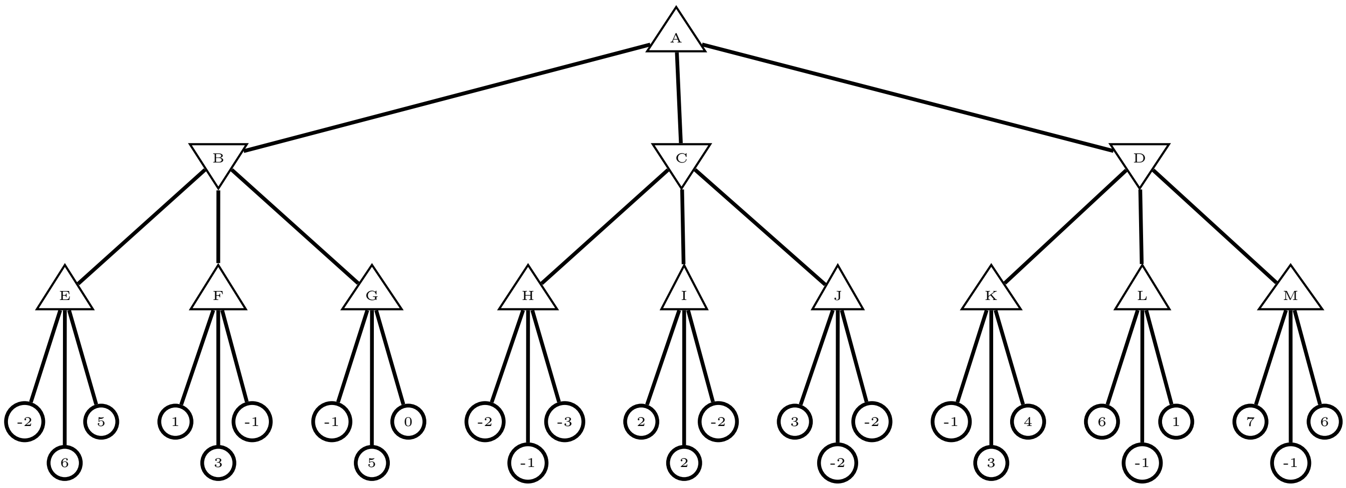
- Indicate with dark edges what move should be chosen by the players at each MAX or MIN node.

  `See the edges in the diagram above.`

Should you play this game with Minnie? Explain why or why not.

```
On average, you will expect to lose $1 every time you play, so it
would not be a rational decision to play the game.
```

4. **(10 marks)** Consider the following game tree, where A, E, F, G, H, I, J, K, L, M are MAX nodes, and where B, C and D are MIN nodes.



(a) Use the minimax algorithm and the alpha-beta pruning algorithm to explore this game tree. For each node A-M, show the sequence of values that are assigned to the node's minimax value, α and β, in a table such as this:

⋮

X:	minimax value:	$\cancel{-\infty}$ 2
	α:	$\cancel{-\infty}$ 2
	β:	$\cancel{\infty}$ $\cancel{8}$ 7
Y:	minimax value:	$\cancel{\infty}$ -3
	α:	7
	β:	$\cancel{\infty}$ -3

⋮

`Answer on next page`

`For each node, here are the sequences of values for the node's minimax value,` α `and` β`.`

A:	minimax value:	$\cancel{-\infty}$ $\cancel{3}$ 4
	α:	$\cancel{-\infty}$ $\cancel{3}$ 4
	β:	∞
B:	minimax value:	$\cancel{\infty}$ $\cancel{6}$ 3
	α:	$-\infty$
	β:	$\cancel{\infty}$ $\cancel{6}$ 3
C:	minimax value:	$\cancel{\infty}$ -1
	α:	3
	β:	∞
D:	minimax value:	$\cancel{\infty}$ 4
	α:	3
	β:	$\cancel{\infty}$ 4
E:	minimax value:	$\cancel{-\infty}$ $\cancel{-2}$ 6
	α:	$\cancel{-\infty}$ $\cancel{-2}$ 6
	β:	∞
F:	minimax value:	$\cancel{-\infty}$ $\cancel{1}$ 3
	α:	$\cancel{-\infty}$ $\cancel{1}$ 3
	β:	6
G:	minimax value:	$\cancel{-\infty}$ $\cancel{-1}$ 5
	α:	$\cancel{-\infty}$ -1
	β:	3
H:	minimax value:	$\cancel{-\infty}$ $\cancel{-2}$ -1
	α:	3
	β:	∞
I:	minimax value:	
	α:	
	β:	
J:	minimax value:	
	α:	
	β:	
K:	minimax value:	$\cancel{-\infty}$ $\cancel{-1}$ $\cancel{3}$ 4
	α:	$\cancel{3}$ 4
	β:	∞
L:	minimax value:	$\cancel{-\infty}$ 6
	α:	3
	β:	4
M:	minimax value:	$\cancel{-\infty}$ 7
	α:	3
	β:	4

(b) Which nodes A-M and which children of nodes E-M are never visited because of alpha-beta pruning?

`Nodes I and J (and all of their children) are never visited. The third child of node G and the second and third children of nodes L and M are also pruned.`

(c) What is the best move for MAX at node A?

`The best move for MAX at node A is to choose to go to node D (which has minimax value 4). (MIN would then make the choice to go to node K, where MAX would choose 4 when presented with the choice between -1, 3 and 4.)`

5. **(10 marks)** For this problem, you will use a Monte Carlo simulation to answer some questions.

   - In the game *Pig*, two players compete to see who can be the first to reach a total score of at least 100.
   - On a player's turn, they roll a single 6-sided die.
     - If they roll a 1 ('Pig'), their turn ends immediately, and they score no points for the turn.
     - If they roll any other number (2, 3, 4, 5, 6), then that number is added to their *turn total*, and they must then choose to *roll again* or to *hold*. If they *hold*, their turn total is added to their overall score, and their turn ends.
   - **Examples:**
     - Anne rolls a 6, then rolls a 3, then rolls a 4, then rolls a 5, and then decides to hold. Her turn total of 18 is added to her overall score.
     - Bob rolls a 5, then rolls a 2, then rolls a 6, and then rolls a 1. As soon as he rolls a 1, his turn ends, with nothing added to his score.
     - Anne rolls a 5, then rolls a 6, and then decides to hold. Her turn total of 11 is added to her overall score, which brings it to 29.
     - Bob rolls a 1 on his very first roll, and his turn ends immediately (with no score).
   - A common strategy for a single turn in this game is a "hold at 20" policy. Using this strategy, a player will continue to roll while their turn total is less than 20. As soon as their turn total becomes greater than or equal to 20, they will choose to hold.

Your task:

(a) In a programming language of your choice, write code that implements **one turn** using the "hold at 20" policy. Repeatedly generate a random number from 1 to 6 until you either roll a 1 (and therefore earn a turn total of 0) or your turn total becomes greater than or equal to 20. (Note that the only possible turn totals for a hold-at-20 policy will be 0, 20, 21, 22, 23, 24, or 25.)

(b) Set up your code so that you can run any number of simulations of a single turn and keep track of the observed probability of earning each of the possible turn totals.

(c) For 100 simulations, for $10,000$ simulations and for $1,000,000$ simulations, write down your observed probabilities (to 5 decimal places) for each of the possible turn totals.

The answers found during my simulations can be found below. For 100 simulations, there will be quite a lot of variance, so student results might not be extremely close to mine. However, for $1,000,000$ simulations, the results should be quite close.

100 simulations:

```
0 0.69000
20 0.09000
21 0.08000
22 0.05000
23 0.07000
24 0.01000
25 0.01000
```

$10,000$ simulations:

```
0 0.61610
20 0.09800
21 0.10100
22 0.07630
23 0.05340
24 0.03800
25 0.01720
```

$1,000,000$ simulations:

```
0 0.62420
20 0.09993
21 0.09488
22 0.07456
23 0.05411
24 0.03524
25 0.01710
```

(d) Using your probability results for $1,000,000$ simulations, what is the **expected value** of your score for a single turn if you use the "hold at 20" policy?

Using the results above for $1,000,000$ simulations, the expected value for a single turn would be:
$0.62420(0) + 0.09993(20) + 0.09488(21) + 0.07456(22) + 0.05411(23) + 0.03524(24) + 0.01710(25) = 8.14879$

(e) Modify your code so that you can test different hold values (hold at 15, hold at 16, *etc.*). What did you find the expected values to be for each of the hold values between 15 and 25, inclusive? Does 20 appear to be the best choice?

```
Again, student results will not match these exactly, but should be
relatively close.

Hold at 15:  7.85663
Hold at 16:  7.94051
Hold at 17:  8.04383
Hold at 18:  8.08008
Hold at 19:  8.10448
Hold at 20:  8.14879
Hold at 21:  8.13792
Hold at 22:  8.12622
Hold at 23:  8.10320
Hold at 24:  8.06821
Hold at 25:  7.96153

With these results, the hold-at-20 strategy appears to be best.
However, depending on the simulation results, some students might
have found that hold-at-19 or hold-at-21 would be slightly better.
```

(f) Submit your answers to questions (c), (d) and (e) on paper, but also submit your code through Desire2Learn.