Name: ______________________________

Student ID: ______________________________

CSE 473 Autumn 2013: Midterm Exam

(closed book, closed notes except for 1-page summary)

Total: 100 points, 5 questions. Time: 50 minutes

Instructions:

1. Write your name and student ID on the first sheet (once you start, write your last name on all sheets). Write your answers in the space provided. If you need more space or scratch paper, you can get additional sheets from the instructor. Make sure you write down the question number and your name on any additional sheets.
2. Read all questions carefully before answering them. Feel free to come to the front to ask for clarifications.
3. *Hint 1*: You may answer the questions in any order, so if you find that you're having trouble with one of them, move on to another one.
4. *Hint 2*: If you don't know the answer to a question, don't omit it - do the best you can! You may still get partial credit for whatever you wrote down. Good luck!

Do not start until you are given the "go-ahead signal"...

Name: ______________________________
Student ID: ______________________________

1) (20 points, 10 each) Agents and Environments

a) Give a PEAS description of the task environment for the following agents:
   i) An autonomous Metro bus in downtown Seattle
   ii) A program that detects faces in a given image
b) For each of the agents above, characterize the environment according to whether it is fully or partially observable, deterministic or stochastic, episodic or sequential, static or dynamic, discrete or continuous, and single or multiagent.

a) Any reasonable answers receive points

b) (i) partially observable, stochastic, sequential, dynamic, continuous, multiagent

(ii) fully observable, deterministic, episodic, static, discrete, single agent

Name: ______________________________

Student ID: ______________________________

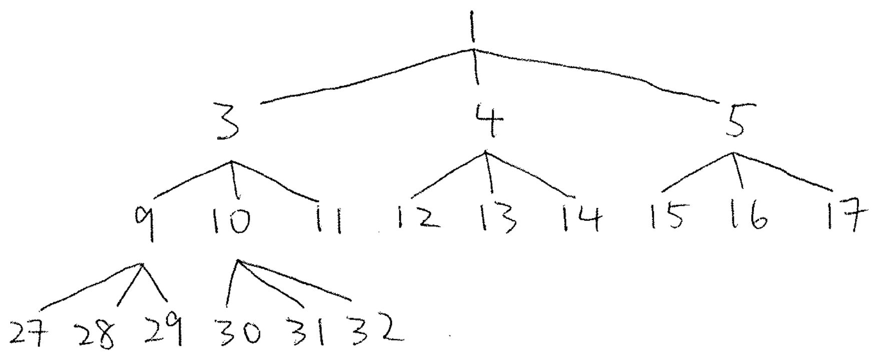
2) (20 points, 5 for a, 15 for b) Uninformed Search

Consider a state space where the start state is 1 and the successor function for any state x returns three states: $3x$, $3x+1$, $3x+2$.

a) Draw the state space graph for states 1 to 32.

b) Suppose the goal state is 30. List the order in which nodes will be visited for: (i) breadth-first search, (ii) depth-limited search with depth limit 2, and (iii) iterative deepening search.

a)



b) (i)

1, 3, 4, 5, 9, 10, 11, 12, 13, 14, 15, 16, 17, 27, 28, 29, 30

(ii) 1, 3, 9, 10, 11, 4, 12, 13, 14, 5, 15, 16, 17

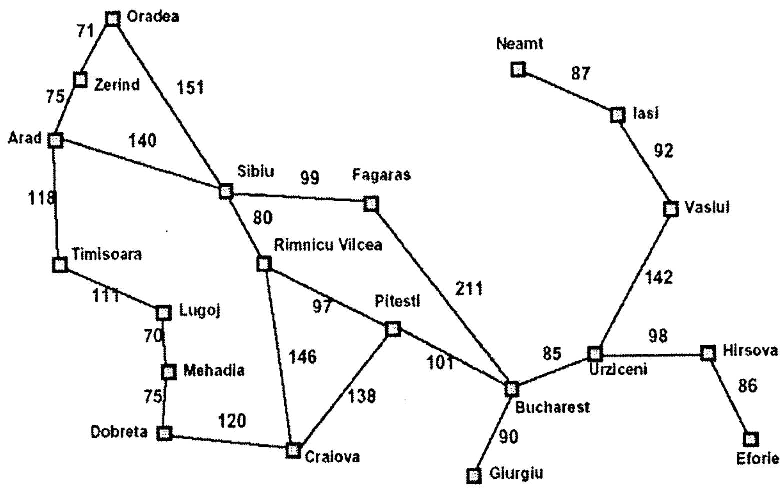
(iii) 1, 1, 3, 4, 5, 1, 3, 9, 10, 11, 4, 12, 13, 14, 5, 15, 16, 17, 1, 3, 9,

27, 28, 29, 10, 30

Name: ____________________
Student ID: ____________________

3) (20 points, 10 each) Informed Search

The map below shows the routes through various towns in Romania. The route distances are labeled on the edges between towns. The table on the right gives the straight line distances to Bucharest from the other towns.



Straight-line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Consider the route-finding problem of computing the cheapest path from **Oradea** to **Bucharest**. For the straight-line distance heuristic, draw the search tree after expansion of each node until the termination of the algorithm for:

a) Greedy best-first search (label all nodes with their h values). What is the solution (list of visited cities) found by the algorithm?

b) A* search (label all nodes with their f values). What is the solution (list of visited cities) found by the algorithm?

Use this page and the next for your answers.

a)

- Oradea $h=380$
  - Zerind $h=374$
  - Sibiu $h=253$
    - Oradea $h=380$
    - Arad $h=366$
    - Fagaras $h=176$
      - Sibiu $h=253$
      - Bucharest $h=0$
    - Rimnicu Vilcea $h=193$

Oradea → Sibiu → Fagaras → Bucharest

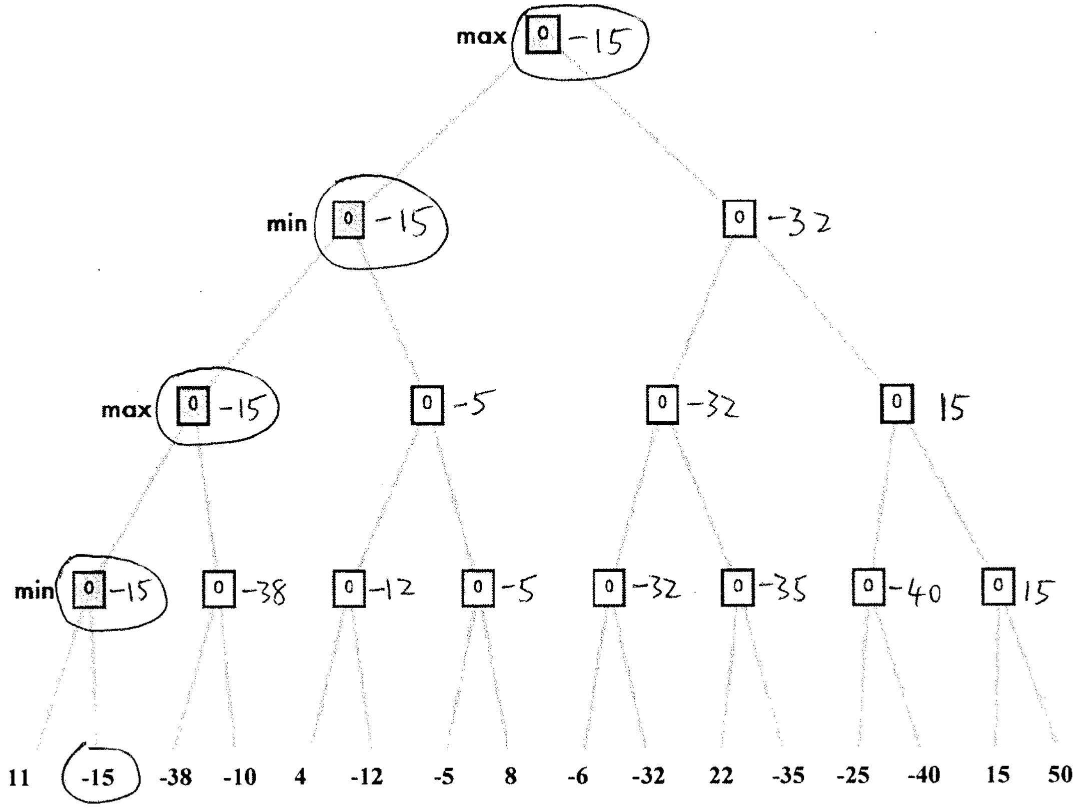
b)

- Oradea $f=380$
  - Zerind $f=445$
  - Sibiu $f=404$
    - Oradea $f=682$
    - Arad $f=657$
    - Fagaras $f=426$
      - Sibiu $f=602$
      - Bucharest $f=461$
    - Rimnicu Vilcea $f=424$
      - Sibiu $f=564$
      - Craiova $f=537$
      - Pitesti $f=428$
        - Rimnicu Vilcea $f=618$
        - Craiova $f=626$
        - Bucharest $f=429$

Oradea → Sibiu → Rimnicu Vilcea → Pitesti → Bucharest

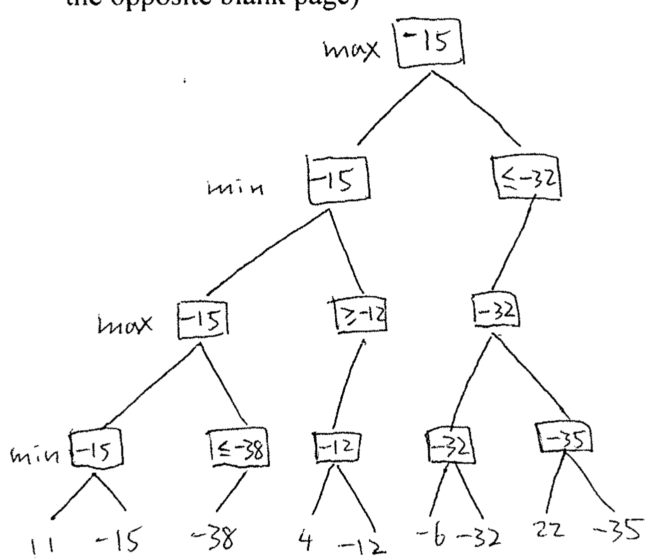
Name: ______________________
Student ID: ______________________

4) (15 points, 5 each) Adversarial Search



Consider the 4-ply game tree above for a two-person game.

a) Fill in the minimax values of all nodes marked 0 in the tree above.
b) Label the path in the tree that results if max and min each make optimal moves.
c) Draw the tree that results if alpha-beta pruning is used. (Use the space below or the opposite blank page)



Name: ______________________

Student ID: ______________________

5) (25 points, 10 for a, 15 for b) Propositional and First-Order Logic

a) State whether each of the following sentences in propositional logic is valid, satisfiable but not valid, or unsatisfiable. Explain your answer.

i) $A \vee (B \Rightarrow B)$

valid

A	B	B⇒B	A∨(B⇒B)
T	F	T	T
F	T	T	T
T	T	T	T
F	F	T	T

→ these are all possibility

ii) $A \wedge \neg B \wedge (A \Rightarrow C)$

satisfiable but not valid

Observe that

A=T, B=F, C=T ⇒ True

A=T, B=T, C=T ⇒ False

b) Suppose a knowledge base contains the following sentences in first-order logic.

$\forall x\ \forall y\ (A(x) \wedge C(x,y)) \Rightarrow \neg B(y)$

$\forall x\ \forall y\ [\neg C(x,y) \Rightarrow D(x)]$

$\exists x\ B(x)$

$A(\text{diddy})$

(where diddy is a constant symbol)

i) Convert the sentences above to CNF (remember to standardize apart the clauses).

ii) Using your CNF sentences in (i), construct a resolution proof that $D(\text{diddy})$.

[Feel free to use the next page if you need more space for your answer]

$\neg D(\text{diddy})$ $\quad \neg A(x_1) \vee \neg C(x_1, y_1) \vee \neg B(y_1)$ $\quad C(x_2, y_2) \vee D(x_2)$ $\quad B(c)$ $\quad A(\text{diddy})$

$\neg C(\text{diddy}, y_1) \vee \neg B(y_1)$

$\neg C(\text{diddy}, c)$

$D(\text{diddy})$

$\perp$