THE UNIVERSITY OF AUCKLAND

FIRST SEMESTER, 2002
Campus: City

COMPUTER SCIENCE

Foundations of Artificial Intelligence

(Time allowed: ONE AND A HALF hours)

NOTE: Closed book.
No calculators.
Attempt all questions.
Put the answers in the boxes below the questions.

MARKS	
HANS:	(out of 65)
MIKE:	(out of 35)
TOTAL:	(out of 100)

SURNAME:

FORENAME(S):

STUDENT ID:

UNIVERSITY UPI:

CONTINUED

UNIVERSITY UPI: ____________________

Section A: Agents

1. Name two characteristics of agents. [2 marks]

2. Name a type of agent (other than “simple reflex agent”). [2 marks]

3. Name two features (other than “accessible vs. inaccessible”) that should be considered when assessing the environment of an agent. [2 marks]

UNIVERSITY UPI: ______________________________

Section B: Prolog

4. Consider the following Prolog program:

```
successor(2,1). successor(3,2). successor(4,3). successor(5,4).
successor(6,5). successor(7,6). successor(8,7). successor(9,8).

larger_digit(X,Y) :- successor(X,Y).
larger_digit(X,Z) :- successor(Y,Z), larger_digit(X,Y).
```

(a) What would be Prolog's answer to the query `successor(2,3)`. [2 marks]

(b) What would be Prolog's answer to the query `successor(3,2)`. [2 marks]

(c) What would be Prolog's answer to the query `larger_digit(7,3)`. [3 marks]

(d) What would be Prolog's answer to the query `larger_digit(3,7)`. [3 marks]

(e) List Prolog's answer to the query `larger_digit(X,5)` in the order that they are produced. [5 marks]

(f) List Prolog's answer to the query `larger_digit(5,Y)` in the order that they are produced. [7 marks]

UNIVERSITY UPI: ____________________

5. What does the following Prolog program do when queried with a constant as first argument, a list as second argument, and a variable as third argument? [5 marks]

```
mystery(X,[X|L],L).
mystery(X,[Y|L1],[Y|L2]) :- mystery(X,L1,L2).
```

6. Write a Prolog fact `third_element/2` that is true if and only if the first argument is the third element in the list given by the second argument. [3 marks]

7. Name the type of search that is performed by the following Prolog program: [3 marks]

```
search(N,[N]) :- goal(N).
search(N,[N|S]) :- successor(N,M), search(M,S).
```

UNIVERSITY UPI: ____________________

Section C: Search

8. Name four informed search strategies. [4 marks]

9. Which search strategy minimizes the cost of the path from the start to the current node? [2 marks]

10. Which search strategy minimizes the estimated cost of the path from the current node to the goal? [2 marks]

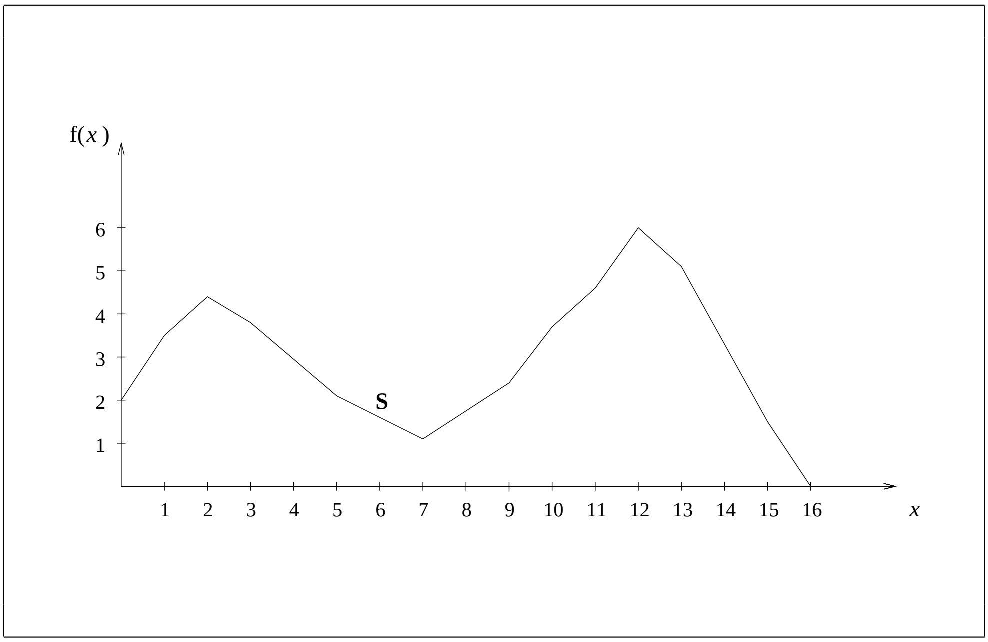
11. Which search strategy minimizes the estimated cost of the path from the start through the current node to the goal? [2 marks]

12. Name two heuristic that can be used to improve a basic backtracking algorithm for constraint satisfaction. [4 marks]

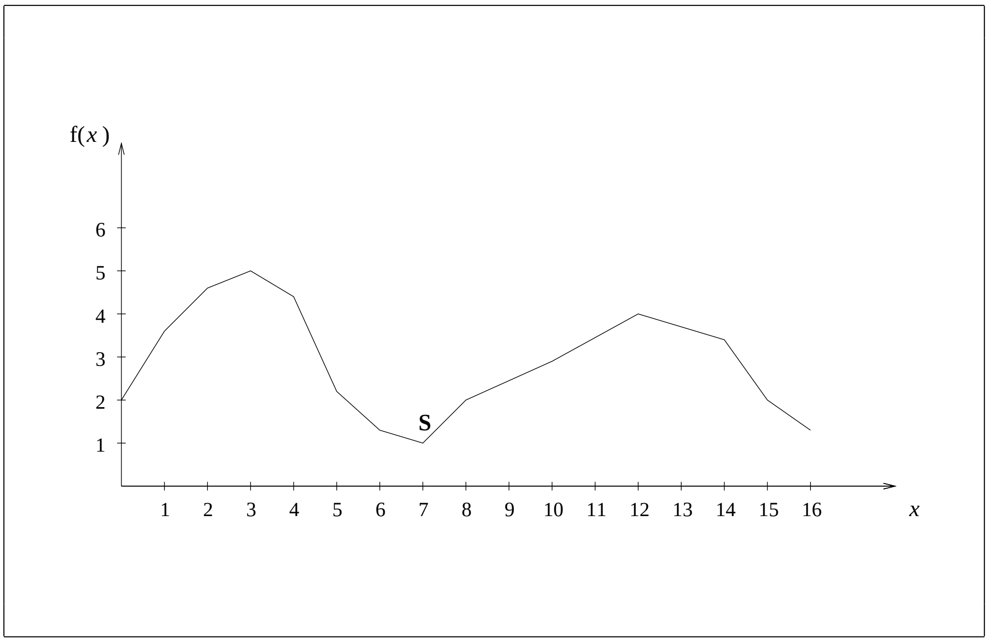
UNIVERSITY UPI: ______________________________

13. Each of the following graphs shows a one-dimensional search space in which the states are represented by integers. The quality of a state x is denoted by $\mathrm{f}(x)$). Each state x has two successor states: $x - 1$ and $x + 1$. Apply steepest-ascent hill climbing to the state indicated with an **S** and mark the final state with an **F**. [8 marks]

(a)

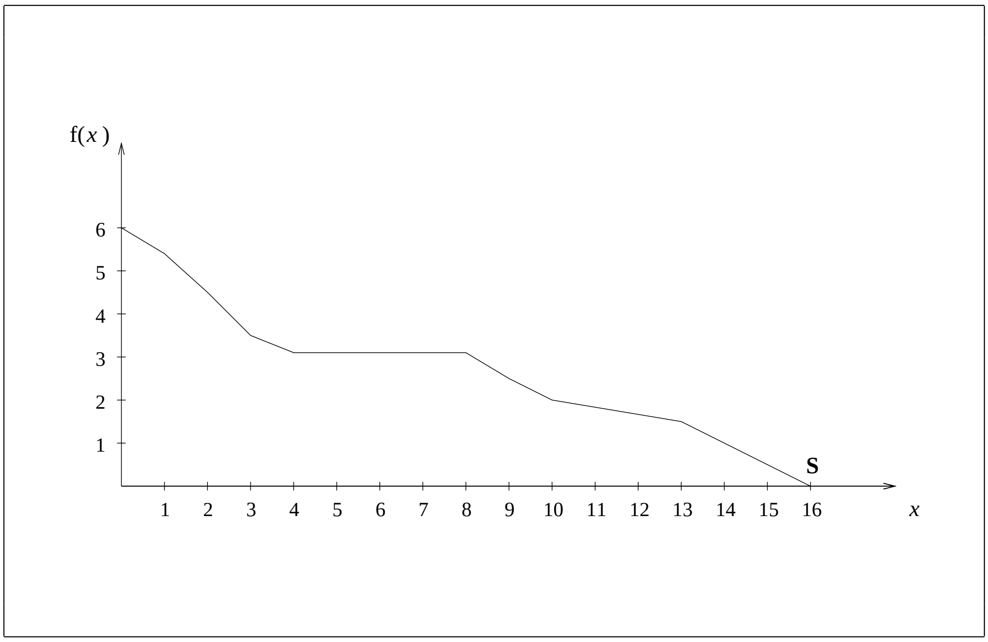


(b)

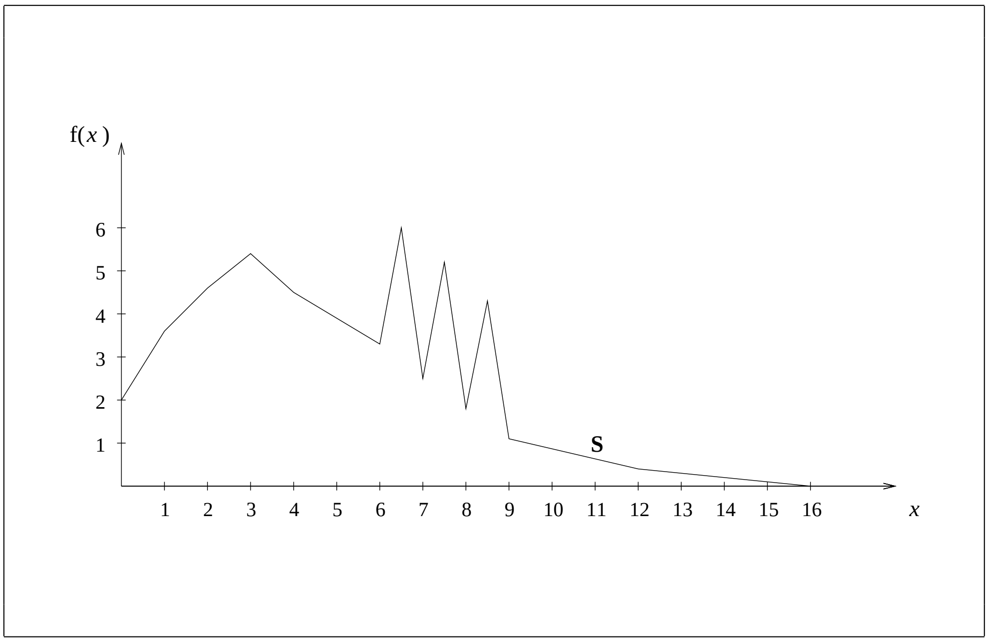


(c)

UNIVERSITY UPI: ______________________________



(d)



UNIVERSITY UPI: ____________________

14. Which of the following statements are true?

    (a) Simulated annealing never accepts a node that is worse than the current node.
    (b) Simulated annealing sometimes accepts a worse node, but the propability of doing so decreases over time.
    (c) Simulated annealing sometimes accepts a worse node, but the propability of doing so increases over time.
    (d) Simulated annealing sometimes accepts a worse node, but this is more unlikely for nodes that are much more worse than for nodes that are only slightly worse.
    (e) Simulated annealing avoids getting stuck in a local optimum by making a big jump from time to time.
    (f) Simulated annealing avoids getting stuck in a local optimum by applying backtracking if there is no better successor node.

[4 marks]

UNIVERSITY UPI: ____________________

Section D: Planning

15. POP Pseudo-Steps [10 marks]

The start and finish pseudo steps were introduced in partially-ordered plans so that the problem's initial situation and top-level goals could be handled the same way as were operator preconditions and effects. For the most part this works, however, there is one important time when a pseudo-step cannot be handled just like an ordinary operator-step.

(a) For which pseudo-step is this true? [3 marks]

(b) When is it true? [3 marks]

(c) Why is it true? [4 marks]

16. POP Threats [15 marks]

Plans and operators are represented here just as they were for assignment 2. Partially-ordered plans are: plan(Steps, CausalLinks, Orderings). Assume you were given the following operator schemas:

```
op1:params=[A,B]                  op2:params=[A,B]
    preconds=[]                       preconds=[m(A,B)]
    effects=[m(A,B)]                  effects=[n(A,B)]

op3:params=[A,B]                  op4:params=[A,B]
    preconds=[]                       preconds=[not(m(A,B))]
    effects=[not(m(A,B))]             effects=[q(B,A)]
```

The following is a plan structure with all information involving the start and finish pseudo-steps omitted.

```
plan(steps([step(p,op1,[a,b]),step(c,op2,[a,b]),step(s1,op3,[a,b]),
            step(s2,op4,[a,b]),step(s3,op4,[a,b]),step(s4,op3,[a,U]),
            step(s5,op3,[V,b]),step(s6,op3,[W,X]),step(s7,op4,[Y,Z]),
            step(s8,op3,[a,b])]),
     causalLinks([causalLink(p,m(a,b),c)]),
     orderings([isBefore(s5,p), isBefore(p,s1), isBefore(s4,s1),
                isBefore(s1,s8), isBefore(s6,s2), isBefore(s8,s2).
                isBefore(s8,c), isBefore(c,s3), isBefore(s3,s7),
                isBefore(p,c)])  )
```

(a) The steps of the plan are shown below with the ordering constraint between step p and step c shown. Add the rest of the rest of the ordering constraints listed above to the drawing. [5 marks]

```
                        S1:not(m(a,b))

              not(m(a,b)):S2

              not(m(a,b)):S3

                        S4:not(m(a,U))

 p:m(a,b)------------------------------------------>m(a,b):c

                        S5:not(m(V,b))

                        S6:not(m(W,X))

              not(m(Y,Z)):S7

                        S8:not(m(a,b))
```

UNIVERSITY UPI: ____________________

Sometimes a step threatens to "clobber" a causal link. A necessary threat to a causal link is one where the step must "clobber" the causal link and a possible threat is one where it might "clobber" the causal link.

(b) What are the names of the steps that are necessary threats to the causal link between step p and c? [5 marks]

(c) What are the names of the steps that are possible but not necessary threats to the causal link? [5 marks]

17. Differences between progession and regression planning. [10 marks]

In the following diagrams, c1(x) and c2(y) are positive object-level conditions. The first diagram shows a progressive planner matching an operator's preconditions against a situation description. The second one shows a regressive planner matching an operator's effects against a goal set.

Progressive Planner Matching Preconditions Against Situation Description:	Regressive Planner Matching Effects Against Goal Set Description:

```
c1(x)    c1(A)+----+            +----+ c1(A)    c1(x)
              | op |            | op |
c2(y)    c2(B)+----+            +----+ c2(B)    c2(y)
```

(a) What are the possible sets of bindings for A and B generated by the progressive planner's matching procedure. [5 marks]

(b) What are the possible sets of bindings for A and B generated by the regressive planner's matching procedure? [5 marks]

UNIVERSITY UPI: ______________________________

Rough Working (NOT MARKED)

UNIVERSITY UPI: ____________________

Rough Working (NOT MARKED)