1. **List the order in which nodes are visited in the tree below for each of the following four search strategies.**
   1. Depth-first search (choosing left most branches first)

1 2 5 6 10 11 3 7 12 13 4 8 9

* 1. Iterative deepening search (increasing the depth by 1 each iteration and choosing left most branches first)

1 1 2 3 4 1 2 5 6 3 7 4 8 9 1 2 5 6 10 11 3 7 12 13 4 8 9

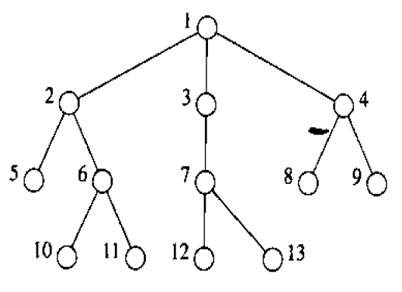
* 1. Breadth-first search

1 2 3 4 5 6 7 8 9 10 11 12 13

* 1. Uniform cost search. Assume that the cost from the root to each node is as follows:

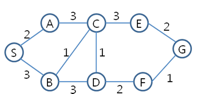
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| node | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| cost | 0 | 2 | 1 | 3 | 4 | 3 | 5 | 10 | 5 | 6 | 9 | 7 | 10 |

1 3 2 4 6 5 7 9 10 12 11 8 13



1. Consider the following search space. *S* is the start state and *G* is the goal state. In the table, *h(X)* is the heuristic cost from *X* to the goal.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Node | S | A | B | C | D | E | F | G |
| *h(X)* | 6 | 4 | 4 | 4 | 5 | 1 | 1 | 0 |



* 1. Is *h(X)* admissible? Describe the reason.

h(X)가 admissible하기 위해서는 모든 X에 대해서 h(X) <= h\*(X)를 만족해야 한다. h\*(X)는 X부터 goal까지의 실제 cost를 말한다. 하지만 위의 예제의 경우 h(D) = 5인데 h\*(D) = 3이다. 따라서 admissible하지 않다.

* 1. Is *h(X)* consistent? Describe the reason

h(X)가 consistent하기 위해서는 모든 노드 u, v에 대해서 h(u) – h(v) <= cost(u,v) 를 만족해야 한다. Cost(u,v)는 노드 u와 노드 v 사이의 cost를 말한다. 하지만 위의 예제에서 h(D) – h(F) = 4인데 cost(D,F) = 2이다. 따라서 consistent하지 않다.

* 1. Show the path found by the A\* algorithm. Initially, there is only one node *S* in OPEN. At the first iteration, *S* is expanded and its children are inserted into OPEN.

|  |  |  |
| --- | --- | --- |
|  | Node  expanded | Open list |
| 0 |  | S=0+6 |
| 1 | S | A=2+4, B=3+4 |
| 2 | A | B=3+4, C=5+4 |
| 3 | B | C=5+4, C=4+4, D=6+5 |
| 4 | C | C=5+4, D=6+5, D=5+5, E=7+1 |
| 5 | E | C=5+4, D=6+5, D=5+5, G=9+0 |
| 6 | C revisited | D=6+5, D=5+5, G=9+0 |
| 7 | G |  |
| 8 |  |  |
| 9 |  |  |
| 10 |  |  |

1. Consider the following cost functions for A\* search. In all cases, *g(n)* is the cost of a node *n* from the start node, *h(n)* is an admissible heuristic cost from *n* to the goal. Assume all costs are positive. You will use one of the following cost functions as *Coste(n)*. Which cost functions guarantee the optimal solution?

D, g

* 1. *g(n)*
  2. *2\*g(n)*
  3. *h(n)*
  4. *g(n)+h(n)*
  5. *2\*g(n) +h(n)*
  6. *g(n)+2\*h(n)*