Q1-1: What are disadvantages of IDA* search?

1. IDA* has no disadvantages
2. IDA* sometimes returns a suboptimal solution
3. IDA* can visit the same state multiple times during the same iteration
4. When IDA* restarts, it discards all information except the next threshold
5. 3 and 4

Q1-1: What are disadvantages of IDA* search?

1. IDA* has no disadvantages
2. IDA* sometimes returns a suboptimal solution
3. IDA* can visit the same state multiple times during the same iteration
4. When IDA* restarts, it discards all information except the next threshold
5. 3 and 4

Q1-2: Consider the Beam search example with *k* = 2 and what would happen if we increased *k* to 3. Under what conditions will Beam search return the optimal solution?

1. When no states are evicted from OPEN because the capacity *k* has been reached

2. When the same solution is returned for *k* and *k*+1

3. When the heuristic used is admissible

Q1-2: Consider the Beam search example with $k = 2$ and what would happen if we increased k to 3. Under what conditions will Beam search return the optimal solution?

1. When no states are evicted from OPEN because the capacity k has been reached

2. When the same solution is returned for k and $k+1$

3. When the heuristic used is admissible

Q1-3: Consider proof that A* is optimal. Why can we not use this proof by contradiction to prove that Beam search is optimal?

1. Beam search may not find any path to the goal
2. If Beam search finds a suboptimal path to the goal, it may not have stored unexpanded node *n* on the optimal path
3. Beam search no longer guarantees f(n)=g(n)+h(n)
4. 1 and 2
5. 2 and 3

- Suppose A* finds a suboptimal path ending in goal *G'*, where $f(G') > f^*$ = cost of optimal path
- Let's look at the first unexpanded node *n* on the optimal path (*n* exists, otherwise the optimal goal would have been found)
- $f(n) \geq f(G')$, otherwise we would have expanded *n*
- $f(n) = g(n)+h(n)$

  $= g^*(n)+h(n)$

  $\leq g^*(n)+h^*(n)$

  $= f^*$
- $f^* \geq f(n) \geq f(G')$, contradicting the assumption at top

Q1-3: Consider proof that A* is optimal. Why can we not use this proof by contradiction to prove that Beam search is optimal?

1. Beam search may not find any path to the goal
2. If Beam search finds a suboptimal path to the goal, it may not have stored unexpanded node *n* on the optimal path
3. Beam search no longer guarantees f(n)=g(n)+h(n)
4. 1 and 2
5. 2 and 3

- Suppose A* finds a suboptimal path ending in goal G', where $f(G') > f^*$ = cost of optimal path
- Let's look at the first unexpanded node n on the optimal path (n exists, otherwise the optimal goal would have been found)
- $f(n) \geq f(G')$, otherwise we would have expanded n
- $f(n) = g(n)+h(n)$

  $= g^*(n)+h(n)$

  $\leq g^*(n)+h^*(n)$

  $= f^*$
- $f^* \geq f(n) \geq f(G')$, contradicting the assumption at top

Q2-1: Which of these is ***NOT*** a reason to prefer optimization instead of search for the 3-SAT optimization problem with a very large number of Boolean variables and clauses.

1. Path cost is not relevant for 3-SAT
2. The state space is very large
3. We need to guarantee we find the Boolean assignments that satisfy the most clauses possible

Q2-1: Which of these is *NOT* a reason to prefer optimization instead of search for the 3-SAT optimization problem with a very large number of Boolean variables and clauses.

1. Path cost is not relevant for 3-SAT
2. The state space is very large
3. We need to guarantee we find the Boolean assignments that satisfy the most clauses possible

Q2-2: You are building a computer that requires *k* parts. You must purchase them from *k* stores because each store has a limit of 1 part per customer. Finding the least expensive way to buy the parts is a:

1. Uninformed search problem

2. Informed search problem

3. Optimization problem

Q2-2: You are building a computer that requires *k* parts. You must purchase them from *k* stores because each store has a limit of 1 part per customer. Finding the least expensive way to buy the parts is a:

1. Uninformed search problem
2. Informed search problem
3. Optimization problem ⟵

Q2-3: You are buying *k* computer parts from *k* stores, as in question 2-2. A state is a mapping of which parts you buy from which stores. What is a good choice of neighbor?

1. Select the most expensive part; enumerate the *k*-1 ways to buy that part from the other stores by swapping parts mapped to those stores

2. For each set of 3 parts and their 3 stores, enumerate all the different ways to buy those parts from the 3 stores

3. Fix the store for 1 part; randomly reassign all other *k*-1 parts

Q2-3: You are buying *k* computer parts from *k* stores, as in question 2-2. A state is a mapping of which parts you buy from which stores. What is a good choice of neighbor?

1. Select the most expensive part; enumerate the *k*-1 ways to buy that part from the other stores by swapping parts mapped to those stores

2. For each set of 3 parts and their 3 stores, enumerate all the different ways to buy those parts from the 3 stores

3. Fix the store for 1 part; randomly reassign all other *k*-1 parts

Q3-1: In the 3-SAT example we started from the all T state. Consider what would happen if we instead started in the all F state.

1. Hill climbing would still find the global optimum with *f* = 2

2. Hill climbing would find the global optimum if it visited neighbors in the right order

3. Hill climbing would get stuck in a local optimum

$A \vee \neg B \vee C$
$\neg A \vee C \vee D$
$B \vee D \vee \neg E$
$\neg C \vee \neg D \vee \neg E$
$\neg A \vee \neg C \vee E$

(Note the recorded discussion of this quiz question incorrectly used AND operators instead of OR).

Q3-1: In the 3-SAT example we started from the all T state. Consider what would happen if we instead started in the all F state.

1. Hill climbing would still find the global optimum with $f = 2$
2. Hill climbing would find the global optimum if it visited neighbors in the right order
3. Hill climbing would get stuck in a local optimum

$A \lor \neg B \lor C$
$\neg A \lor C \lor D$
$B \lor D \lor \neg E$
$\neg C \lor \neg D \lor \neg E$
$\neg A \lor \neg C \lor E$

(Note the recorded discussion of this quiz question incorrectly used AND operators instead of OR).

Q3-2: In the 3-SAT example we reached a local optimum in the state (A=T, B=F, C=T, D=T, E=T). How can we modify hill climbing to escape this local optimum?

1. In "IF $f(t) \leq f(s)$ THEN stop" change $\leq$ to $<$

2. We cannot fix hill climbing to escape this local optimum

3. Pick the neighbor t with the smallest $f(t)$ instead of the largest $f(t)$

$A \vee \neg B \vee C$
$\neg A \vee C \vee D$
$B \vee D \vee \neg E$
$\neg C \vee \neg D \vee \neg E$
$\neg A \vee \neg C \vee E$

(Note the recorded discussion of this quiz question incorrectly used AND operators instead of OR).

Q3-2: In the 3-SAT example we reached a local optimum in the state (A=T, B=F, C=T, D=T, E=T). How can we modify hill climbing to escape this local optimum?

1. In “IF $f(t) \leq f(s)$ THEN stop” change $\leq$ to $<$
2. We cannot fix hill climbing to escape this local optimum
3. Pick the neighbor t with the smallest $f(t)$ instead of the largest $f(t)$

$A \vee \neg B \vee C$
$\neg A \vee C \vee D$
$B \vee D \vee \neg E$
$\neg C \vee \neg D \vee \neg E$
$\neg A \vee \neg C \vee E$

(Note the recorded discussion of this quiz question incorrectly used AND operators instead of OR).

Q3-3: What is a disadvantage of modifying hill climbing to explore a plateau?

1. Unlike the original hill climbing, it will now be sensitive to the initial state

2. It may waste time exploring and never improve the *f* score

3. It may choose a neighbor with a lower score than the current state

Q3-3: What is a disadvantage of modifying hill climbing to explore a plateau?

1. Unlike the original hill climbing, it will now be sensitive to the initial state

2. It may waste time exploring and never improve the *f* score

3. It may choose a neighbor with a lower score than the current state