

Hongwei Liu

CS510 HW2

1.A

1. The branching factor in this state space is 4.
2. There are $4k$ distinct states at depth level k .
3. In a tree search, there would be b^{d+1} states explored so there will be 4^{x+y+1} nodes.
4. For a graph search, since the start point is $(0, 0)$ and the goal is at (x, y) so the depth would be $x+y$, the number of states would be $2(x+y)(x+y+1)+1$.
5. Yes. $|u-x| + |v-y|$ is actually a calculation for Manhattan distance for (u, v) and goal (x, y) which is used as an admissible heuristic in A^* search.
6. For heuristic, the move would be $x+y$ because it is the least steps that heuristic think to move from $(0, 0)$ to (x, y) . Every move will expand nodes around it. Since all the nodes might be on the optimal path to goal, and the heuristic value will affect the next move which change latter the number of nodes expand. So it is hard to know the exact expanded number of nodes.
7. Yes, if some links are removed which almost equal to put on some blocks. Blocks will cost more steps but won't change the heuristic value and the heuristic value is still underestimated.
8. No. Adding new links to nonadjacent states will reduce the actual distance from one point to another point which might make heuristic value higher than the real costs thus heuristic might no longer be underestimated.

1.B

1. Any movements might have negative costs which would give search algorithms chances to drop down the total path cost to the goal. So any optimal uninformed search algorithms would like to explore all the nodes to get as many as negative costs as possible to find an even better optimal (lower cost) path to the goal.
2. For tree search, if we already know the number of nodes and remaining nodes. Assuming the cost of each action is m . The total cost in n paths are $-mn$ which will be larger than $-cn$, the best cost to a goal. so other paths worse than $-cn$ will be ignored.

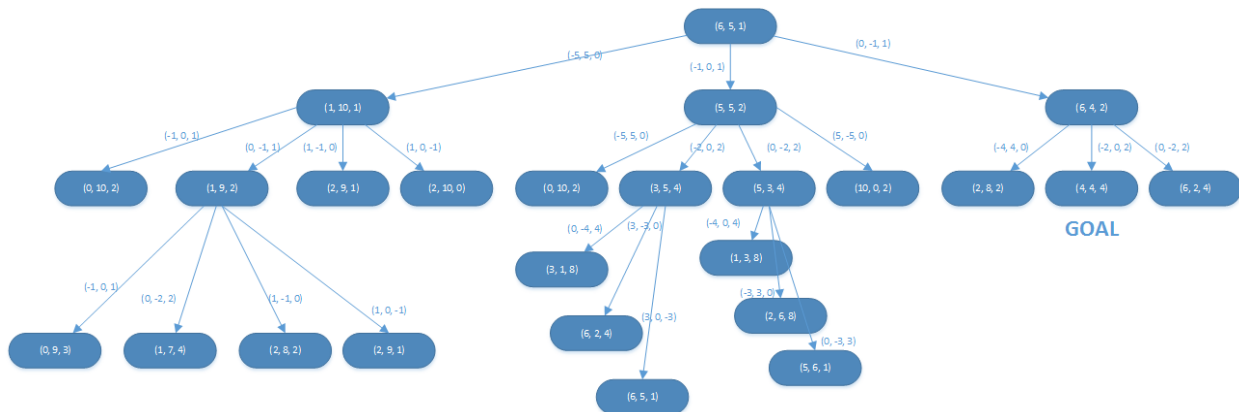
For graph search, it might keep looping for any times to get more negative costs to lower down total cost.
3. The agent will keep looping forever because it keeps finding a lower cost path. However, the agent will go to other paths if those paths have even lower cost.
4. People will get bored if they keep watching the same beautiful sceneries. So people will get less and less value from beautiful sceneries when they keep looping in a scenery visit. In state space search, we should put the already visited scenery into storage and each time we are going to visit a "new" scenery,

we should normalize the scenery to compare with the scenery in storage. For example, a person might feel more excited to go to a rainforest instead of a sea somewhere if he/she already had seen many kinds of sea sceneries before. In addition, the time and fuel spent might play the roles of heuristic that people might prefer to do some domestic travels instead of international ones.

5. Playing video games, having delicious food and traveling around since people are easy to get more negative costs by loop these actions.

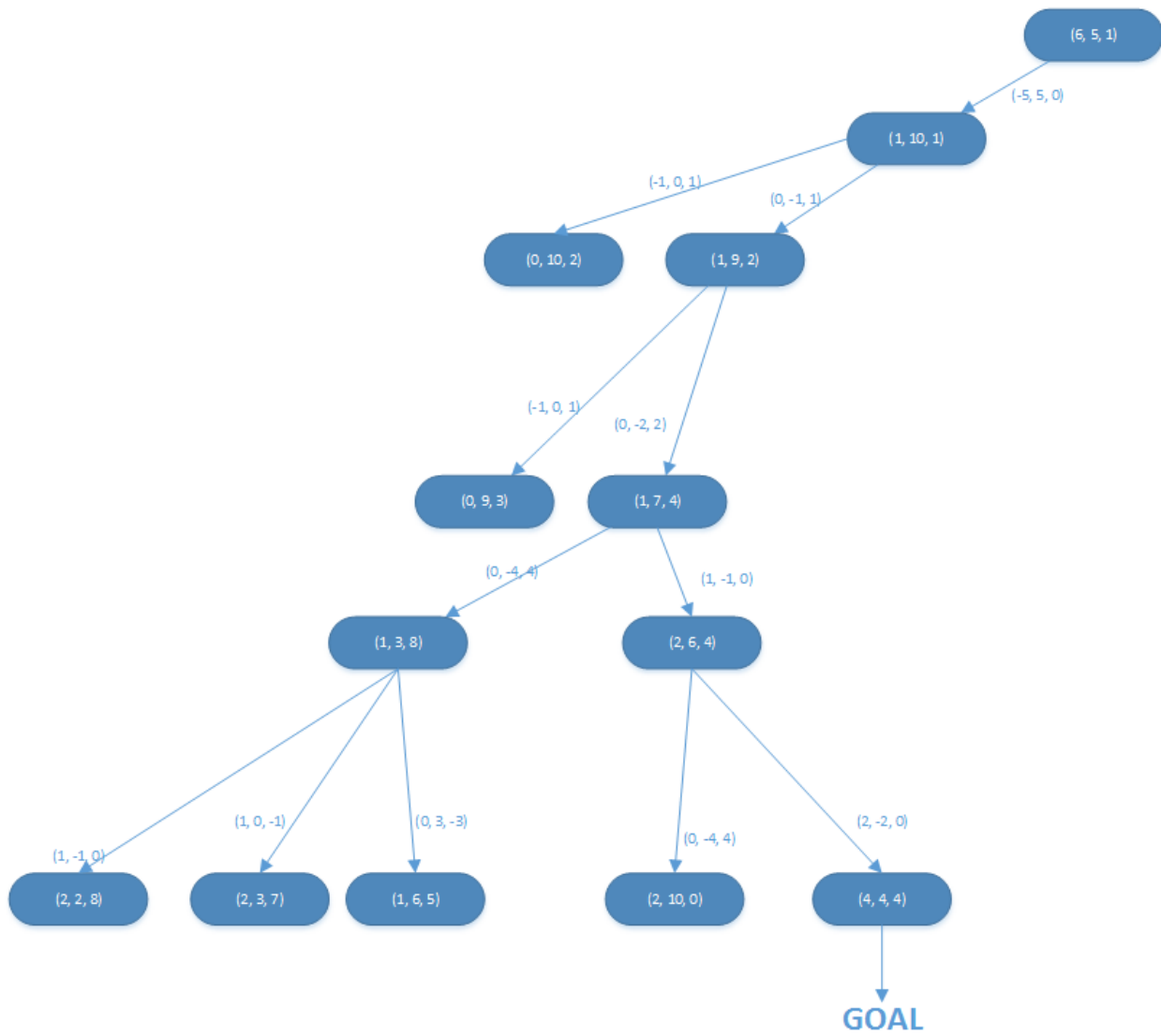
1.C

1.



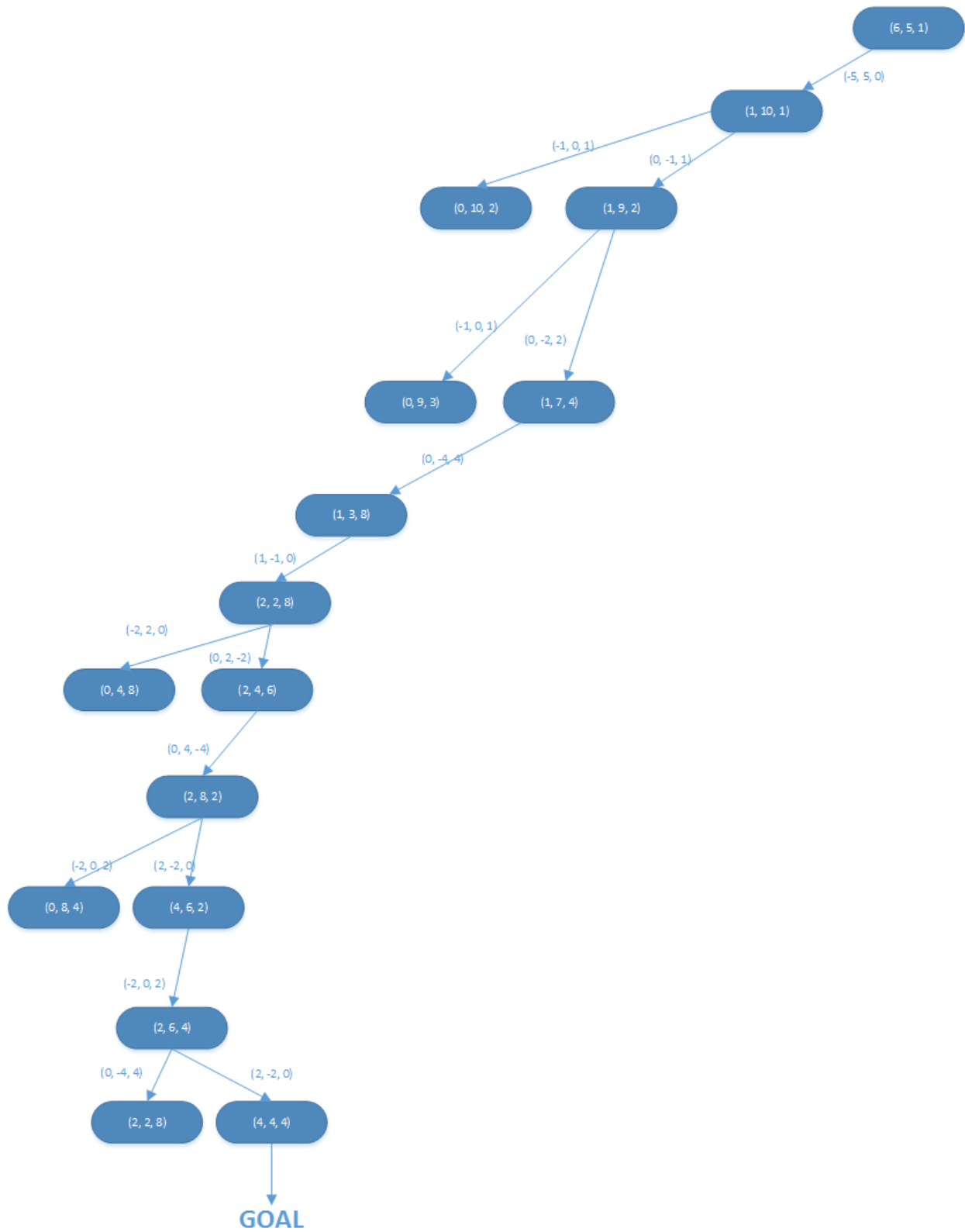
Backtrack: 10

2.



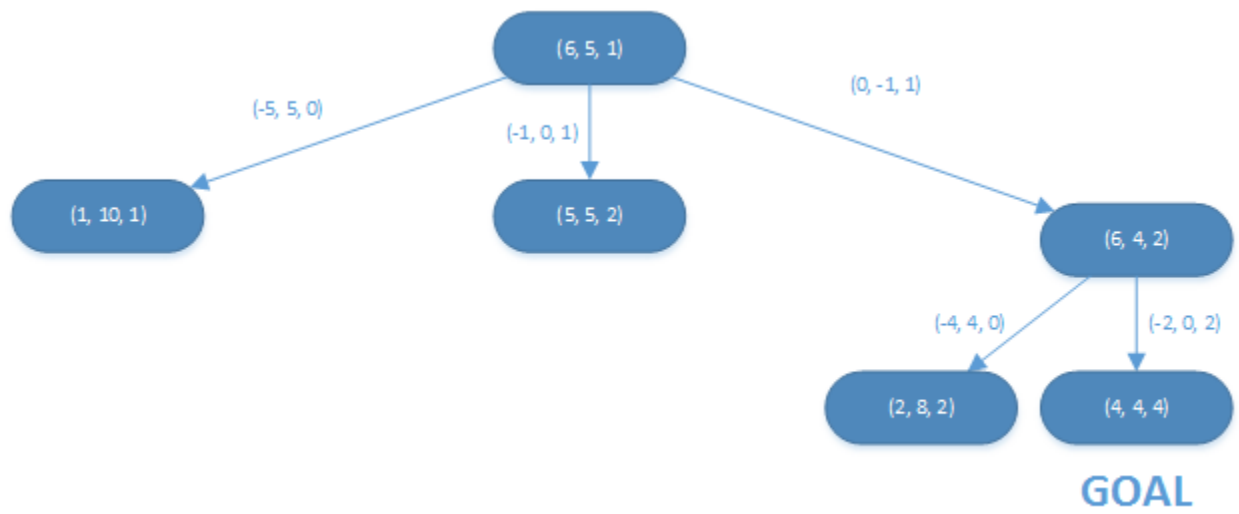
Backtracks: 4

3.



Nodes generated: 15

4.



Nodes generated: 5