Randy Lam

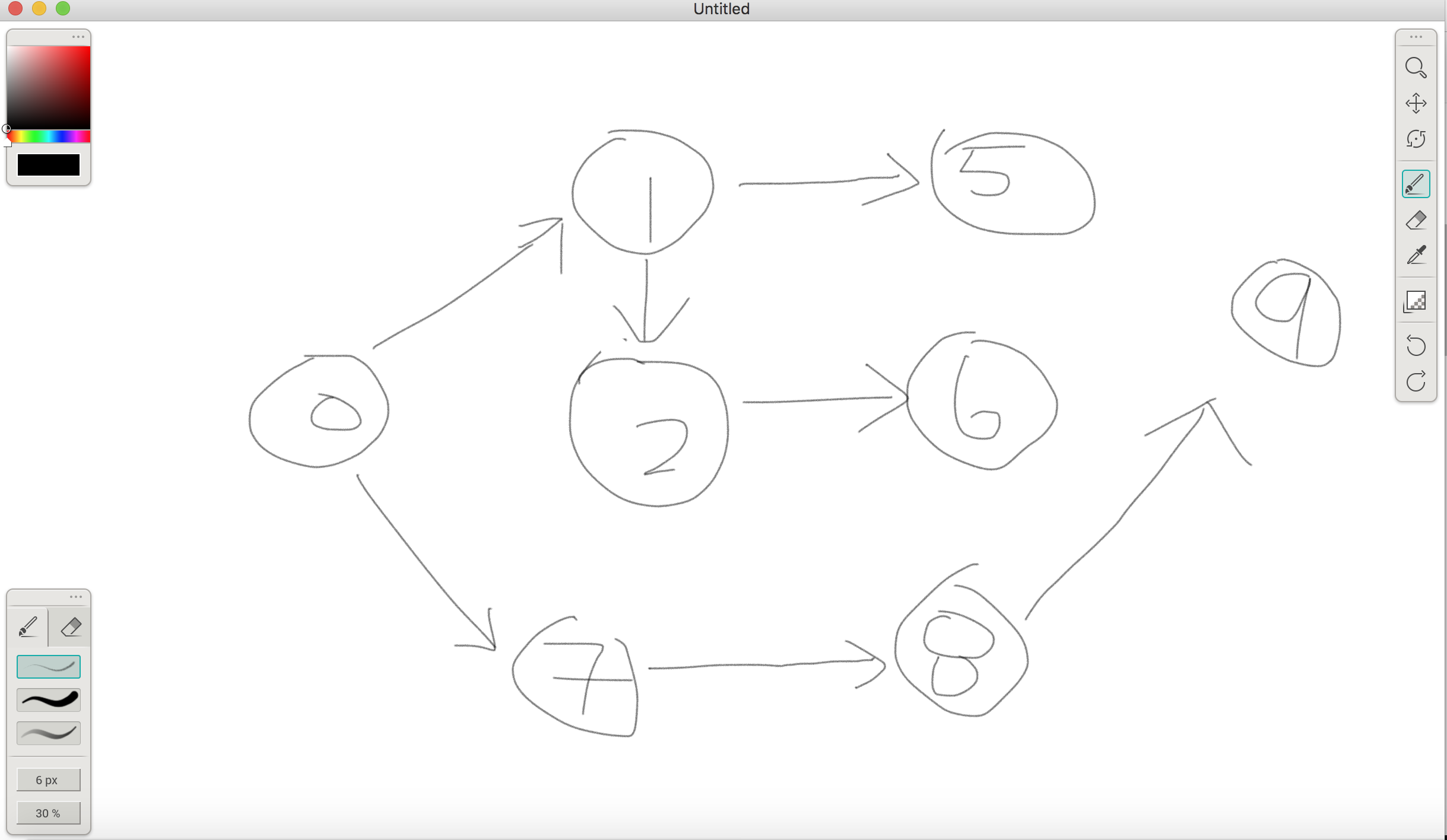
Homework 2 CSE 180

**Dijkstra’s Algorithm**

1.)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Step | OPEN | xS | A | B | C | D | E | F | xg |
| 0 | xS/0 | N/0 | N/∞ | N/∞ | N/∞ | N/∞ | N/∞ | N/∞ | N/∞ |
| 1 | A/1, C/7, B/10 | N/0 | xS/1 | xS/10 | xS/7 | N/∞ | N/∞ | N/∞ | N/∞ |
| 2 | C/7, B/2, D/5 | N/0 | xS/1 | A/2 | xS/7 | A/5 | N/∞ | N/∞ | N/∞ |
| 3 | B/2, D/5, F/8 | N/0 | xS/1 | A/2 | xS/7 | A/5 | N/∞ | C/8 | N/∞ |
| 4 | D/5, E/6,  F/8 | N/0 | xS/1 | A/2 | xS/7 | A/5 | B/6 | C/8 | N/∞ |
| 5 | D/5, E/6, xg/9 | N/0 | xS/1 | A/2 | xS/7 | A/5 | B/6 | C/8 | F/9 |
| 6 | D/5, xg/9 | N/0 | xS/1 | A/2 | xS/7 | A/5 | B/6 | C/8 | F/9 |
| 7 | xg/9 | N/0 | xS/1 | A/2 | xS/7 | A/5 | B/6 | C/8 | F/9 |

2.)

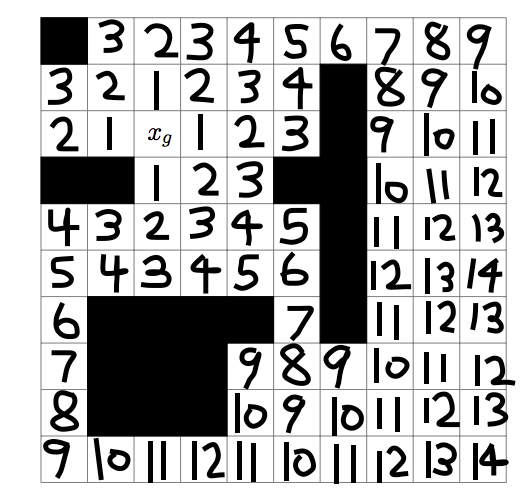


**A\***

1.) A consistent heuristic is admissible if it never overestimates the cost for any given node. Additionally, the table does provide an admissible heuristic for the cost to. Starting from X\_s, the heuristic cost to get to node X\_s is 8. Now to get to all the accessible nodes, A,B, and F, the heuristic cost to go through the nodes have to be higher or equal to the previous node’s heuristic cost. Using the path to get to node A, we compare the estimation of X\_s to get to A. To get to A we add the heuristic estimation for it (7), plus the path to get there (2), plus the cost of the parent which isn’t relevant in this example, this would equal 9 which is greater than the heuristic cost the node before it. Doing this for the other nodes now, to get to path B, the heuristic cost of node B is 5 plus the path to get there so it’s 8 which is again fine as long as it isn’t less than the previous node’s heuristic estimation. Finally going to path F, the cost is 10 compared to 8 which still satisfies the rules. Also because h(xg) = 0, the function is consistent and since all the vertices are less than the previous one, this table provides both an admissible and a consistent heuristic function.

**Navigating Functions 1**

1.)



**Navigating Functions 2**

1.) The values provided within this grid should be a valid function because it still satisfies the three rules which are ψ(xg) = 0, which means that there is a goal point with 0 cost. If x∈V is a vertex from which there is no path to xg, then ψ(x)=∞, which means if there isn’t a path to xg, then the path is infinity but that rule doesn’t matter here, and lastly, if x is a state from which xg can be reached, then for y=e’(x) we have ψ(y) < ψ(x) which means if the goal can be reached, then for the function y=e’(x), the values of ψ(y) and ψ(x) plugged into the equation will always result to ψ(y) < ψ(x). The graph below simply points to a the next lowest path in chronological order to reach to the goal point.

