

Homework 6

Question 1

1. It can print adjacency list of each vertex which has been added to the T, at least I can't any other list.
2. $P5Last = 2/3$. 3 can be added lastly after 5s to T only in one case, because 5 has more connections than 2. $P5Last = 1 - P3Last = 1 - 1/3 = 2/3$.

Question 2

I guess the key point in this algorithm is the random pop method. Since this queue (S) pops elements randomly, there is no difference in which order vertexes will be added to the queue (S).

Question 3

The color property plays role as visited / not visited value for a vertex. Each while loop iteration adds connections to the queue (S) for not visited randomly popped vertex. Once all vertexes visited the queue (S) won't be pushed with new vertexes, but random pop will make the queue (S) empty at some iteration. Graph G is immutable during while loop execution which means the queue goes throughout a limited number of vertexes. At some iteration the while loop will get to the vertex v and will add it to the queue T, since initially vertex v is not visited but there is a connection from v to t.

Question 4

$\Theta(V) = |V|^2$, from practical experience I guess that a random pop operation might be complex enough in order to implement random vertex removal. Or elements in the S must be shuffled first in order to get a random vertex. Some existing algorithms can do it in n time.

Question 5

Maybe there is a catch, but it looks like f position in T looks predictable because vertexes a, b, c, d, e got no children. In this case f is going to be added lastly. No matter in what order a, b, c, d, e will be called, since there is the only one child which is f.