COMP 3270 Assignment 4 (100 points)

**Due by 11:59PM on Monday, November 28th, 2022**

Instructions:

1. Late submissions **will not** be accepted unless prior permission has been granted or there is a valid and verifiable excuse.
2. Think carefully; formulate your answers, and then write them out concisely using English, logic, mathematics and pseudocode (no programming language syntax).
3. Type your final answers in this Word document.
4. Don’t turn in handwritten answers with scribbling, cross-outs, erasures, etc. If an answer is unreadable, it will earn zero points. **Neatly and cleanly handwritten submissions are acceptable**.

**1. (15 points)** Show d and π values that result from running Breadth First Search on the directed graph below using vertex 3 as the start node.

d=

d=

π =

π =

d=

π =

d=

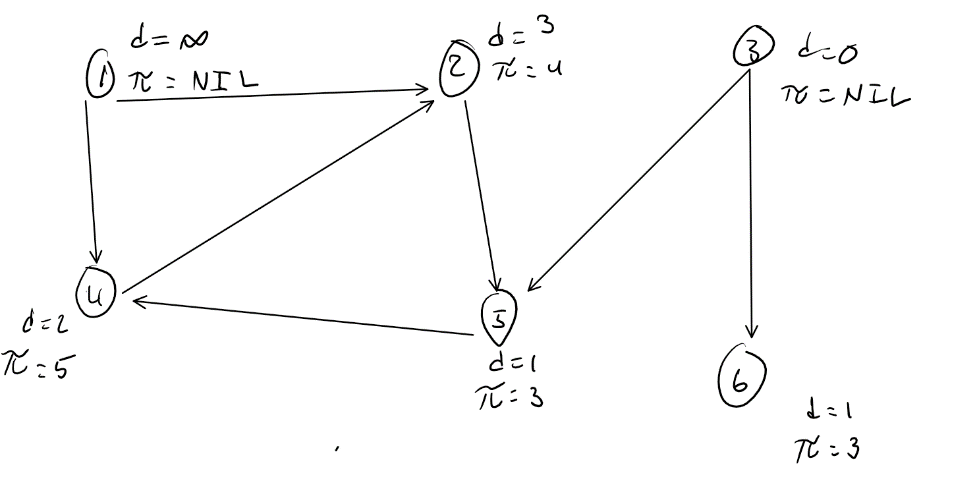
π =

π =

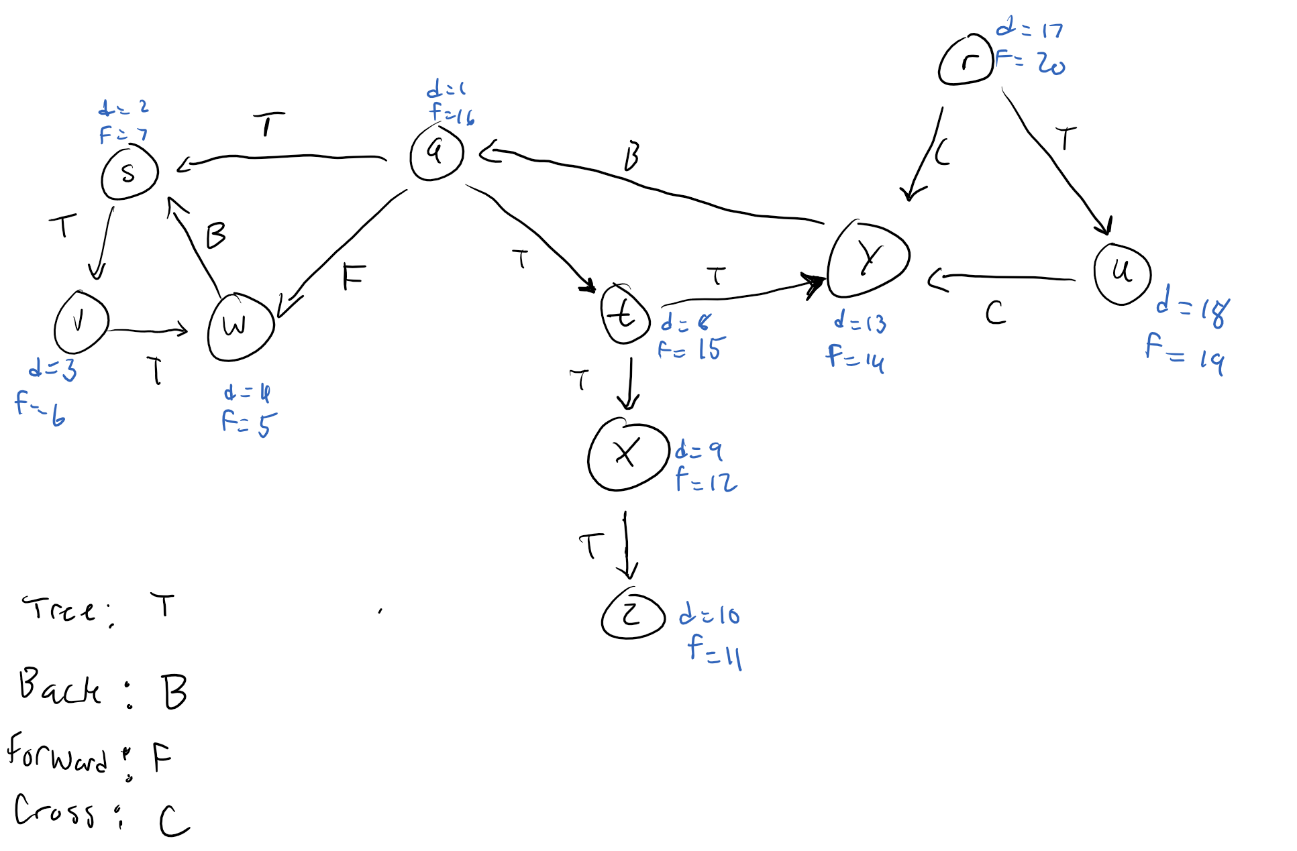
π =

d=

d=

****

**2. (10 points)** Show how Depth First Search works on the graph below by marking on the graph the discovery and finishing times (d and f) for each vertex and the classification of each edge. Assume that the for loops in DFS and DFS-VISIT consider vertices alphabetically.

****

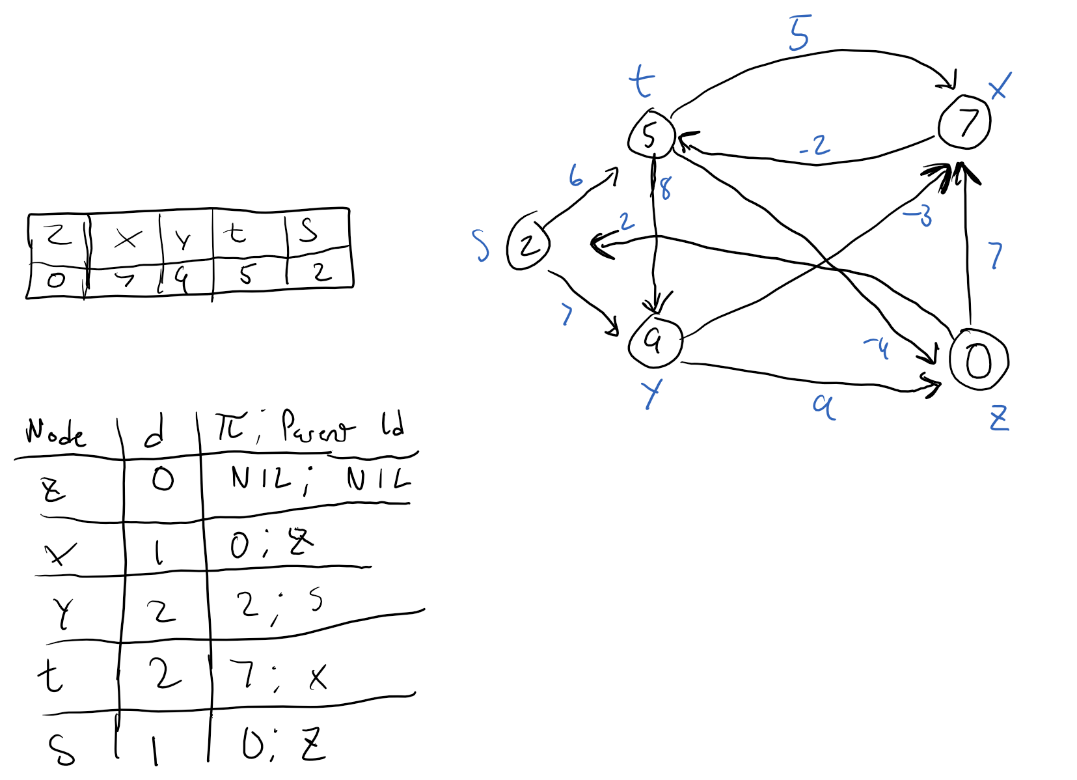
**3. (15 points)** List the vertices of the graph below in Topological Order, as produced by the Topological Sort algorithm. Assume that the for loops in DFS and DFS-VISIT consider vertices alphabetically.

* q,t
* u, q, t
* z, u, q, t
* w, z,u,q,t
* x, w, z, u, q, t
* v, x, y, z, u, q, t
* y, v, x, z, u, q, t
* r, y, v, x, z, u, q, t
* m, r, y, v, x, z, u, q, t
* s, m, r, y, v, x, z, u, q, t
* o, s, m, r, y, v, x, z, u, q, t
* n, o, s, m, r, y, v, x, z, u, q, t
* p, n, o, s, m, r, y, v, x, z, u, q, t

Topological Sort Result:

* p, n, o, s, m, r, y, v, x, w, z, u, q, t

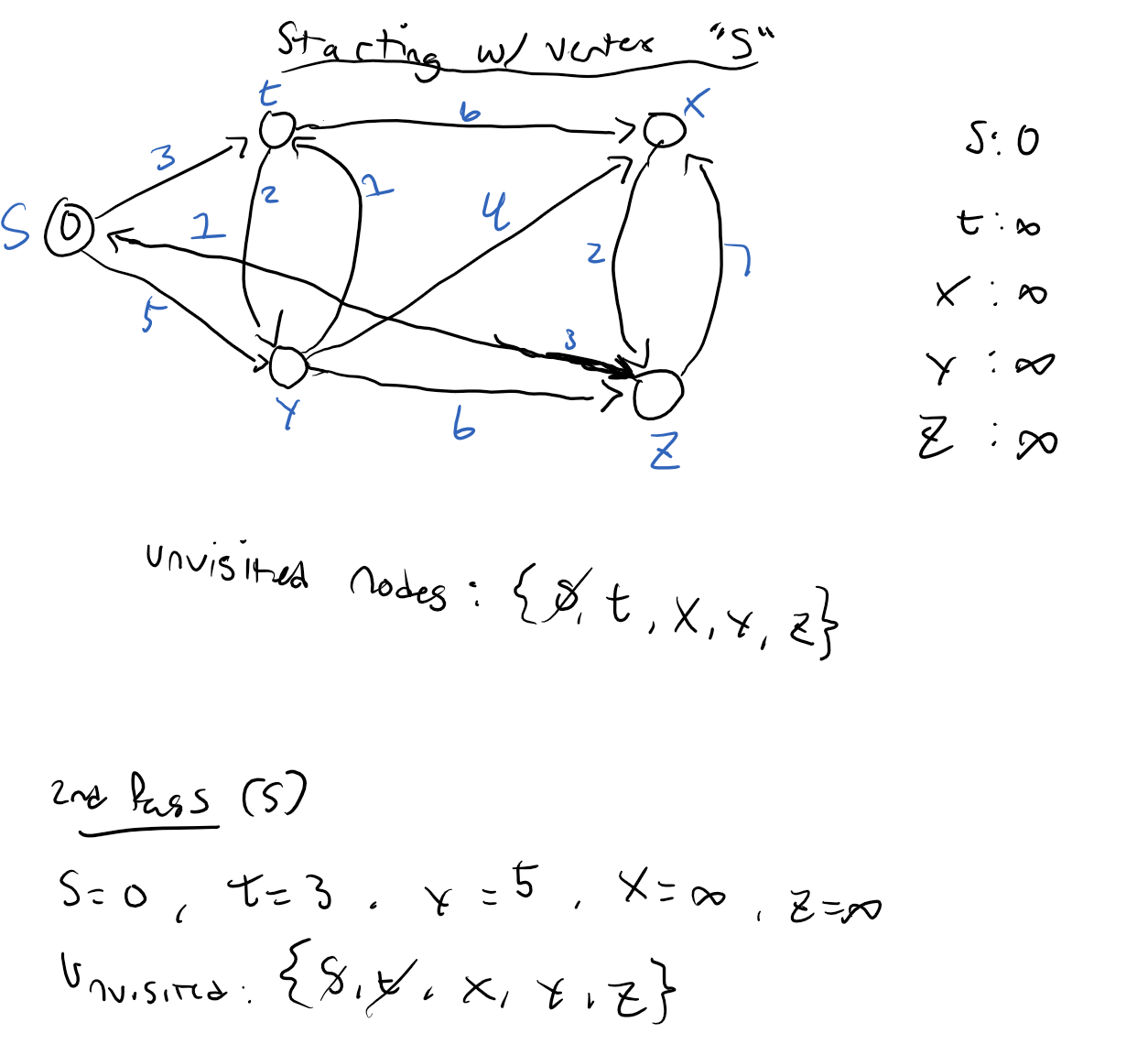
**4. (15 points)** Do Problem 24.1-1 (p. 654) (you do not have to do the last part, i.e., running the algorithm again after changing an edge weight).

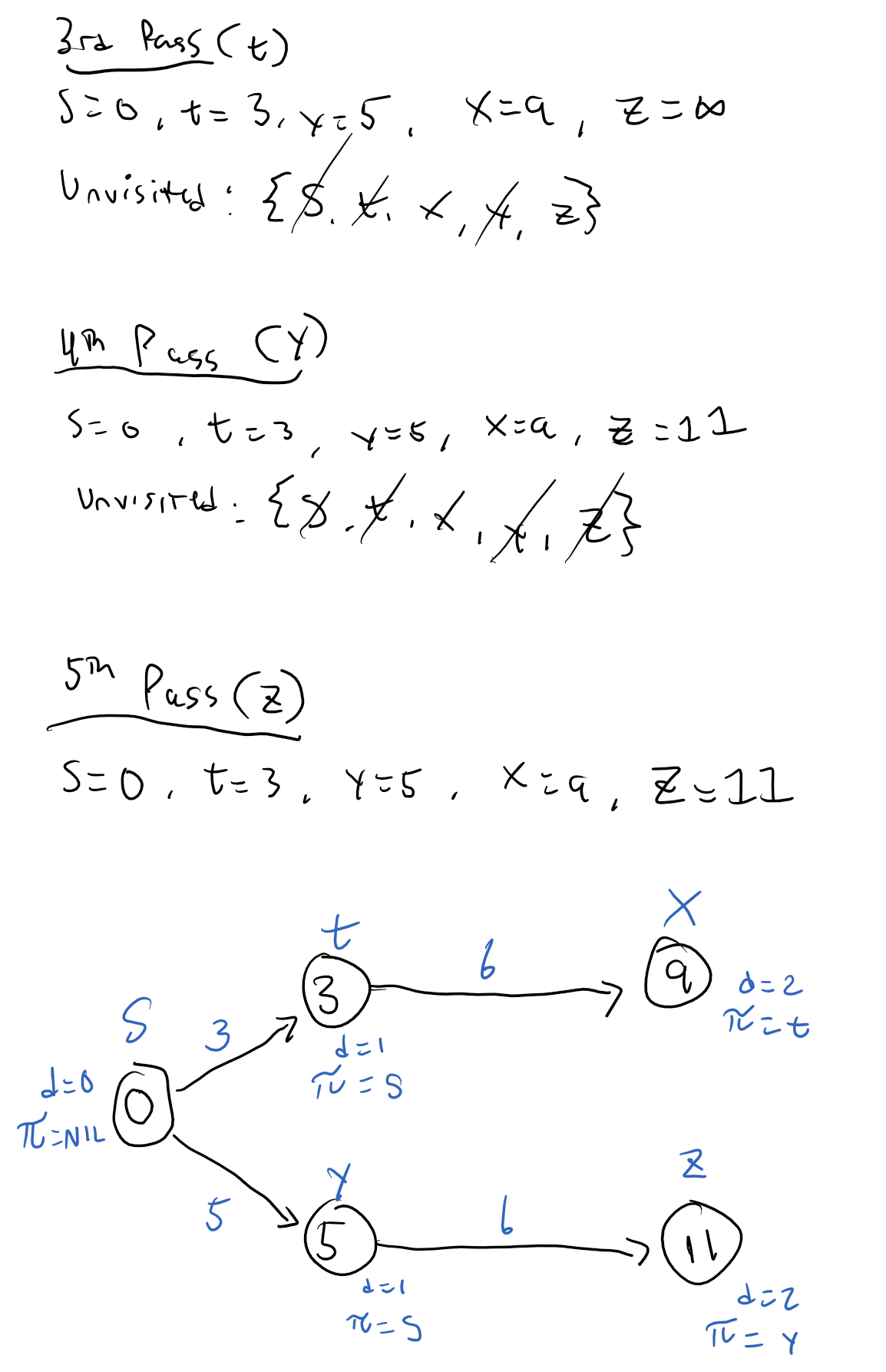


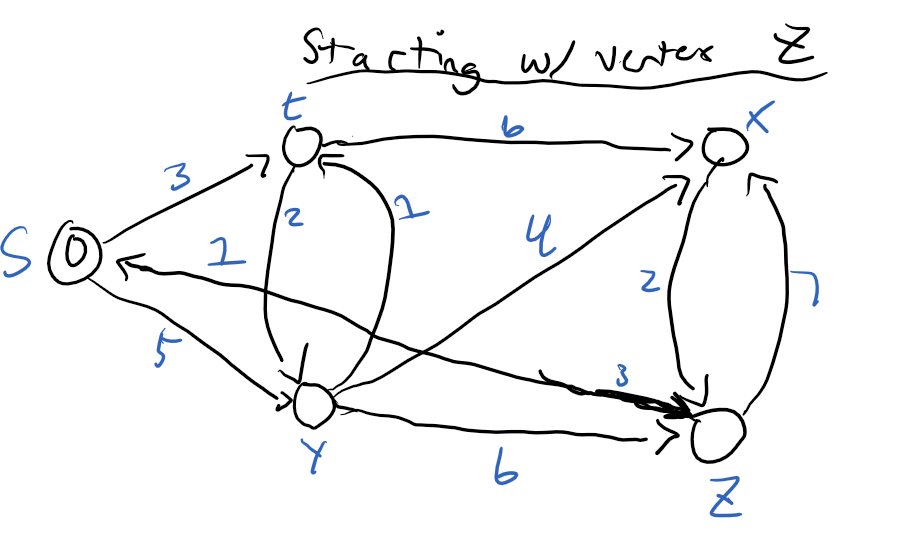
**5. (15 points)** Do Problem 24.2-1 (p. 657 of the recommended text). Show the results similar to Fig. 24.5.

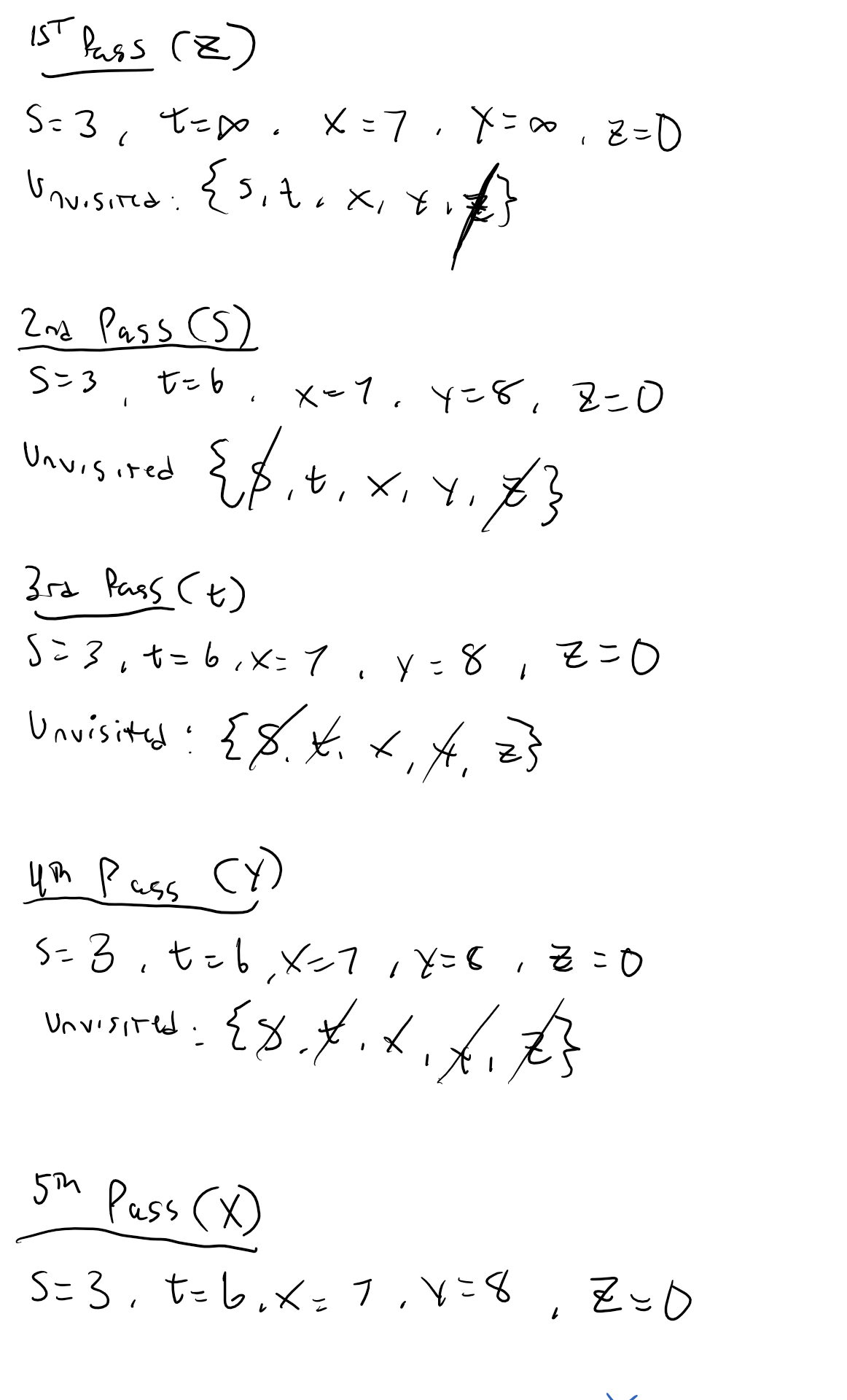
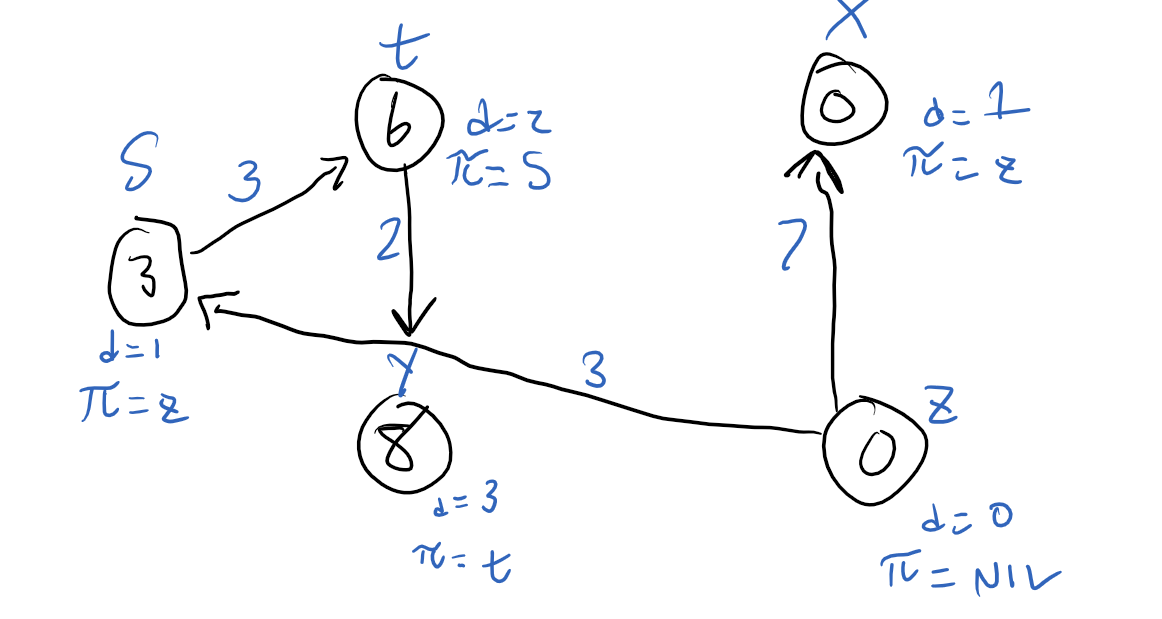


**6. (20 points)** Do Problem 24.3-1 (p. 662 of the recommended text).







**(7) (10 points)** Supposethat a graph G has a Minimum Spanning Tree (MST) computed. How quickly can we update the MST if we add a new vertex and incident edges to G. Propose and outline a strategy and present an algorithm (you can reuse graph algorithms covered in class as building blocks as part of your solution) and evaluate its asymptotic complexity.

* + Whenever you add a new component to a MST, you’re doing the union of 2 MSTs. If you add a new vertex, then it will result in a new minimum spanning tree. If you use a greedy MST strategy, it will help find the shortest path to each vertex in graph G.
  + If you sort the edges by weight, add edges by weight, add edges to the MST from smallest to largest edge, and add edges that do not form a cycle (like Kruskal’s algorithm) then we would achieve a proper MST for computation
  + This would run 0(ElogV) where E contains the incident edges that are added in this process