A: Consider some further analysis of the setting of problem 5.B using the network flow graph suggested (bipartite flow network with processors on one side and lines on the other). To avoid confusion with using \( n \) and \( m \) for the number of vertices and nodes in the flow graph, let \( p \) be the number of processors and \( c \) the number of communication lines. Thus \( n = p + c + 2 \) and \( m = |L_1| + \ldots + |L_p| + p + c \)

i) Suppose we run the Successive Shortest Augmenting Path (SSAP) algorithm on the . What is the run time on this graph (try and give a tight bound)?

ii) argue that we can improve the run time by only using the SSAP algorithm until the distance label of \( s \) gets to \( \sqrt{(n)} \), then switch to the Ford Fulkerson algorithm until we reach the max flow. Give the run time for this better algorithm.

iii) Suppose that \( p << c \). Argue that the longest augmenting path possible is now much shorter than length \( n - 1 = p + c + 1 \). Use this to improve the running time of the SSAP algorithm in this setting.

B: We now consider some improvements of the solution for 5.C where we want the minimum number of days to test the lines (assuming we can’t do it in one day). To test a schedule using \( k \) days we use use capacities of \( k \times t_i \) for the \((s, P_i)\) arc. Call this the \( k - \)day network. Consider the following approach:

We find a max flow in the 1-day network using Ford Fulkerson. If it works, done. Otherwise increase the capacities on the arcs out of \( s \) to be for a 2-day network. Use the prior flow as the starting flow, and run the Ford Fulkerson algorithm again. If this fails to test all lines, update to a 3-day network, and use final flow found in the 2-day network as the starting flow. Continue until you reach the first network which works.

i) argue that this approach works (finds the correct max flow for \( k \)-days, and \( k \) is the minimum number of days which works).

ii) What is the run time of your algorithm?

iii) Describe how to extend this approach to use the SSAP algorithm in each network instead of FF.

iv) combine with the ideas of part A) above to get a very fast algorithm to find the minimum number of days to test all lines.

C: Chapter 6 PP problems 4 and 7 page 65,66.

D: 7.6, problem 2 and 9 page74,75