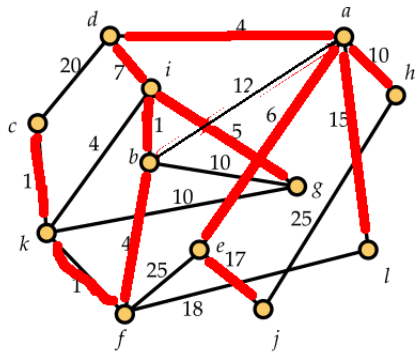
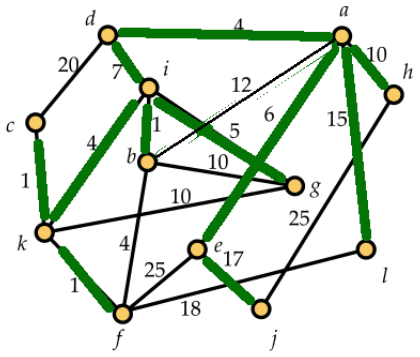
Nithin Raghavan
Math 3012 Combinatorics

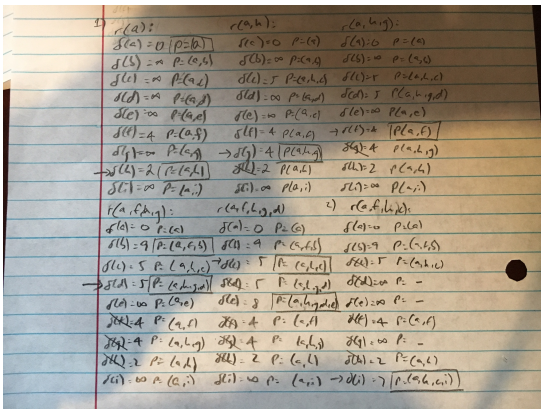
3. The edges must be ordered from smallest to largest, as the smallest weights are preferred. Then, going from the smallest weight (1) to the largest (25), possible edges are considered; if they do not form a cycle, then they are listed, but if they do, then they are ignored, and the next edge is considered. In the course of running this algorithm, the following edges were skipped: (ik), as it leads to the cycle kbfi, (bg) leads to gbi, (kg) leads to kgibf, (fl) leads to balf, (cd) leads to cdabfk, (ef) leads to efba and (jh) leads to jeah. The complete cycle is fk, kc, ib, bf, ad, ig, ea, ah, id, al, ej, and the total weight is $1 + 1 + 1 + 4 + 4 + 5 + 6 + 7 + 10 + 15 + 17 = 71$.



4. Setting the vertex a as root, then the algorithm can proceed by continually choosing the node with the smallest-weighted edge, ignoring cycles. The complete cycle is ad, ae, ah, id, bi, ik, kc, kf, ig, ai, ej. Between kf and ig, (bf) was removed due to it forming a cycle kibf, between ig and ai, (id) was removed for cycle idab, (bg) was removed for cycle gbi, (gk) was removed for cycle gki. After ej, (cd) was removed for cycle cdabik, (hj) was removed for cycle hjac, and (ef) was removed for cycle efkiba. The total weight is $4 + 6 + 10 + 7 + 1 + 4 + 1 + 1 + 5 + 15 + 17 = 71$.

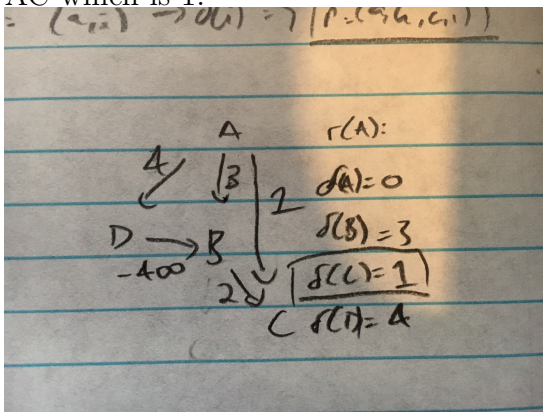


13. $P(a)$ = (a) (0), $P(b)$ = (a,f,b) (9), $P(c)$ = (a,h,c) (5), $P(d)$ = (a,h,g,d) (5), $P(e)$ = (a,h,g,d,e) (8), $P(f)$ = (a,f) (4), $P(g)$ = (a,h,g) (4), $P(h)$ = (a,h) (2), $P(i)$ = (a,h,c,i) (7).

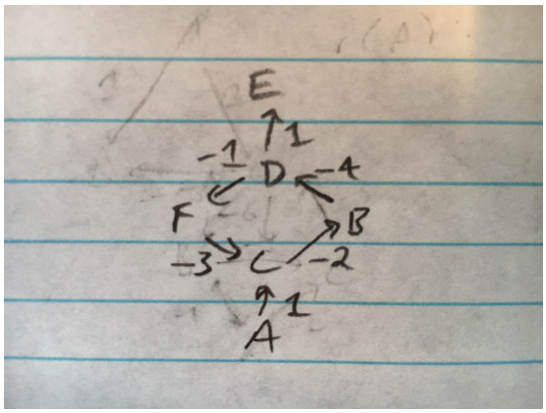


14. The distance from a to a is 0, a to b is 14, a to c is 26, a to d is 10, a to e is 14, a to f is 15. By Dijkstra's algorithm, a path could be {a,d,f,b,e,c}. The shortest paths between the vertices are P(a) = (a), P(b) = (a,b), P(c) = (a,d,c), P(d) = 10, P(e) = (a,d,e) and P(f) = (a,d,f).

16. (a) Djikstra's algorithm relies on the fact that as soon as a shortest path to a vertex is found, then that vertex is "out", and will never have to be reconsidered again, because it is a greedy algorithm. For example, in this graph the shortest path to C is favored. Djikstra's algorithm will look at the path AC and select it, ignoring AD, which although is larger than AC, becomes smaller when DB is considered. So instead of choosing ADBC, which gives a path of -394, it incorrectly chooses AC which is 1.



(b) Consider this example:



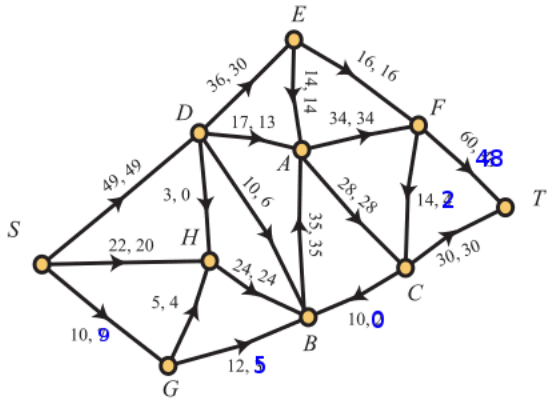
In this case, A is the root and the smallest path to E must be found. The cycle C $\rightarrow$ B $\rightarrow$ D $\rightarrow$ F $\rightarrow$ C should continue on infinitely, after which it can proceed to E with a weight of $-\infty$. But adding 4 to each edge would result in the path A $\rightarrow$ C $\rightarrow$ B $\rightarrow$ D $\rightarrow$ E being taken with a weight of 12.

1. (a) For every vertex in a network diagram besides the source and sink vertex, the net ϕ entering the system must be equivalent to the amount of ϕ leaving the system. For vertex B, the amount entering the vertex is $3 + 2 + 9 = 14$, but the amount leaving B is $0 + 13 = 13$. Additionally, for E, 13 enters the vertex but $10 + 4 = 14$ leaves. Thus, this network is invalid.

   (b) For the edge BE, the flow 13 should be changed to 14. This is still less than the capacity of 15, and so fixes the error.

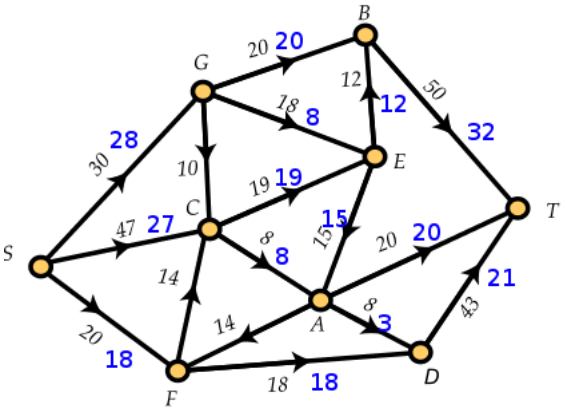
2. Bob is right. Because $\phi \leq C(e)$, then there is no way that the flow could increase more than the capacity of the edges pouring into the sink, which is $6 + 2 + 10 = 18$.

9. The order in which the vertices will be labelled is pseudo-alphabetical, starting with T and then proceeding A, B, C, ... If a node is already labelled, it is ignored, and if it is full on a positive edge or empty on a negative edge, then it is also ignored. An iteration of the Ford-Fulkerson algorithm yields this: S(*,+,∞)', G(S, +, 3)', H(S, +, 2)', B(G, +, 3)', C(B, -, 2)', D(B, -, 3)', F(C, -, 2)', H(D, +, 3), E(D, +, 3), T(F, +, 2). The apostrophes indicate from which vertices scans have taken place, terminating at T. Back-tracking gives an augmented path {S, G, B, C, F, T}. Now, increasing the flows, this can be obtained, giving a maximum flow of 78.



Running the algorithm again to obtain a minimum cut yields S(*, +, ∞), G(S, +, 1), H(S, +, 2), B(G, +, 2), D(B, -, 2), A(D, +, 2), E(D, +, 2) before failing, which gives a minimum cut of L = {S, G, H, B, D, A, E} and R = {F, C, T}.

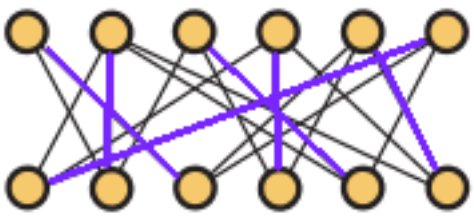
10. Setting all edges equal to zero, and iterating the FF algorithm 7 times, the following graph is obtained:



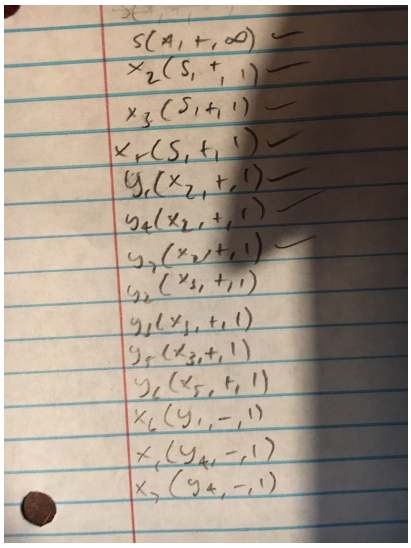
summing the flows that go into T results in a maximum flow of 32 + 20 + 21 = 73, and iterating FF one more time fails, and gives a set, called L, consisting of {S, C, F, G, E}, leaving a set R {A, D, B, T}.

11. Before any conclusions are made, $\phi \leq 75$, or the sums of the capacities of SB, SE and SF. Next, the Ford-Fulkerson algorithm is applied – proceeding in pseudoalphabetical order, the vertices T, A, B, C, D, E and F are considered; A, C, D cannot be formed because they do not border S, and because the second label is F, that means B and E are full, and are ignored in analysis. Thus, if B and E are full, and F can only fit 15, indicating there is an unknown bottleneck, $\phi = 30 + 20 + 15 = 65$.

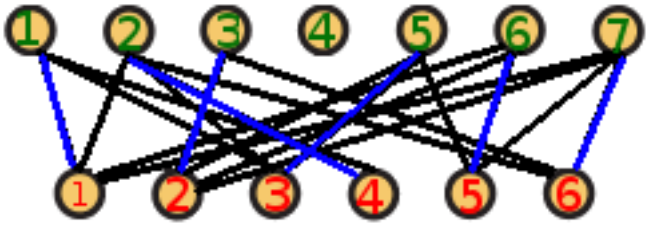
1. Found by inspection:



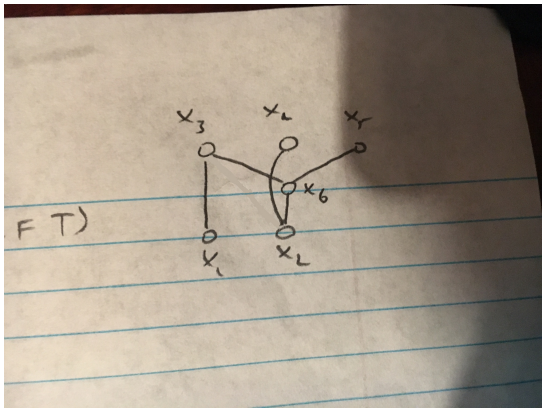
2. A match cannot be made. This is because a select few y-nodes have a high degree compared to the x-nodes, and can be proven using Hall's theorem, where there are defects of nodes that cause d(G) to increase and —X— d(G) to shrink. It can also be shown using the FF algorithm:



4. By assigning the seven schools a number, and assigning the players a number, a bipartite graph can be created, with the following maximum matching by inspection:



5. (a) x_3
   (b) x_2, x_6
   (c) x_3, x_5, x_2
   (d) x_3, x_4, x_5
   (e) x_1, x_2

6. 

7. Because the matching set contains three edges, the poset has a width of three, an antichain of $\{x_3, x_6, x_4\}$ and a partition into w chains of $C_1 = \{x_1, x_3\}$, $C_2 = \{x_2, x_6, x_5\}$, $C_3 = \{x_2, x_4\}$.

8. The width is 3, because there are 3 edges within the matching set, a partition into 3 chains of $C_1 = \{x_4, x_3, x_5\}$, $C_2 = \{x_2, x_6, x_4\}$, $C_3 = \{x_1, x_4, x_6\}$, and an antichain with 3 elements is $\{x_1, x_3, x_2\}$.

9. Because there are 4 edges in the matching set, $w = 4$, and a partition into w chains would include $C_1 = \{x_7, x_5, x_1, x_4\}$, $C_2 = \{x_7, x_10, x_6, x_3, x_4\}$, $C_3 = \{x_2, x_8, x_3, x_4\}$, $C_4 = \{x_2, x_9, x_1, x_4\}$. An antichain would be $\{x_8, x_5, x_9, x_{10}\}$