600.363/463 Solution to Homework 9
Posted: 11/14/03
Due: Wednesday, 11/20/03

26.3-3

By definition, an augmenting path is simple path $s \leadsto t$ in the residual graph G'_f. Since G has no edges between vertices in L and no edges between vertices in R, neither does the flow network G' and hence neith does G'_f. Also, the only edges involving s or t connect s to L and R to t. Note that although edges in G' can go only from L to R, edges in G'_f can also go from R to L.
Thus any augmenting path must go
$s \to L \to R \to ... \to L \to R \to t$
crossing back and forth between L and R at most as many times as it can do so without using a vertex twice. It contains s, t, and equal numbers of distinct vertices from L and R — at most 2+2min($|L|, |R|$) vertices in all. The length of an augmenting path is thus bounded above by 2min($|L|, |R|$)+1.

26-5

1. The capacity of a cut is defined to be the sum of the capacities of the edges crossing it. Since the number of such edge is at most $|E|$, and the capacity of each edge is at most C, the capacity of *any* cut of G is at most $C|E|$.

2. The capacity of an augmenting path is the minimum capacity of any edge on the path, so we are looking for an augmenting path whose edges *all* have capacity at least K. Do a BFS or DFS as usual to find the path, considering only edges with residual capacity at least K (pretend the lower-capacity edges don't exist). This takes $O(V + E) = O(E)$ time. (Note that $|V| = O(E)$ in a flow network.)

3. Max-Flow-By-Scaling uses the Ford-Fulkerson method. It repeatedly augments the flow along an augmenting path until there are no augmenting paths of capacity greater $\geq$1. Since all the capacities are integers, and the capacity of an augmenting path is positive, this means that there are no augmenting paths

whatsoever in the residual graph. Thus, by the max-flow min-cut theorem, Max-Flow-By-Scaling returns a maximum flow.

4. The first time line 4 is executed, the capacity of any edge in G_f equals its capacity in G, and by part (1) the capacity of a minimum cut of G is at most $C|E|$. Intially $K = 2^{\lfloor \lg C \rfloor}$, hence $2K = 2 \cdot 2^{\lfloor \lg C \rfloor} = 2^{\lfloor \lg C \rfloor + 1} > 2^{\lg C} = C$. So the capacity of a minimum cut of G_f is initially less than $2K|E|$.

   - The other times line 4 is executed, K has just been halved, so the capacity of a cut of G_f is at lost $2K|E|$ at line 4 iff that capacity was at most $K|E|$ when the **while** loop of lines 5-6 last terminated. So we want to sow that when line 7 is reached, the capacity of a minimum cut of G_f is most $K|E|$.
     Let G_f be the residual network when line 7 is reached.
     $\neg\exists$ augmenting path of capacity $\geq$ K in G_f
     $\Leftarrow$ max flow f' in G_f has value $|f'| < K|E|$
     $\Leftarrow$ min cut in G_f has capacity $< K|E|$

5. By part (4), when line 4 is reached the capacity of a minimum cut of G_f is at most $2K|E|$, and the maximum flow in G_f is at most $2K|E|$. By an extension of Lemma 26.2, the value of the maximum flow in G equals the value of the current flow in G plus the value of the maximum flow in G_f. Therefore the maximum flow in G is at most $2K|E|$ more than the current flow in G. Every time the inner **while** loop finds an augmenting path of capacity at least K, the flow in G increases by $\geq K$. Since the flow cannot increase by more than $2K|E|$, the loop executes at most $(2K|E|/K = 2|E|$ times.

6. The time complexity is dominated by the loop of lines 4-7. (The lines outside the loop take $O(E)$ time.) The outer **while** loop executes $O(\lg C)$ times, since K is initially $O(C)$ and is halved on each iteration, until $K < 1$. By part (5), the inner **while** loop executes $O(E)$ times for each value of K; and by part (2), each iteration takes $O(E)$ time. So the total time is $O(E^2 \lg C)$.