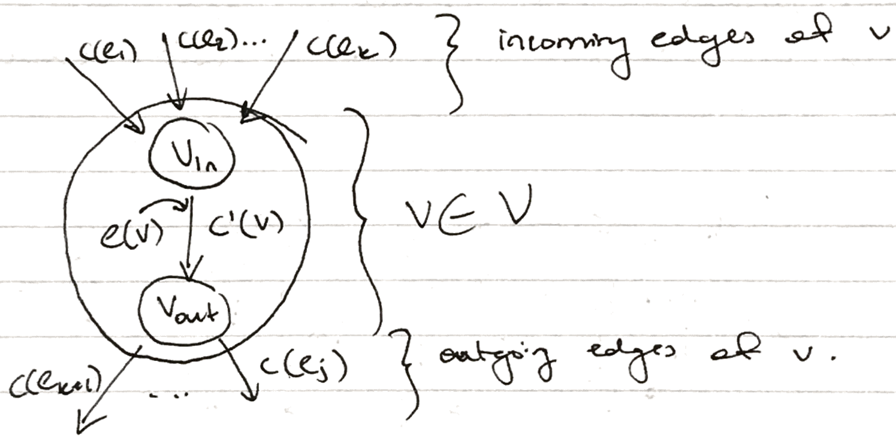
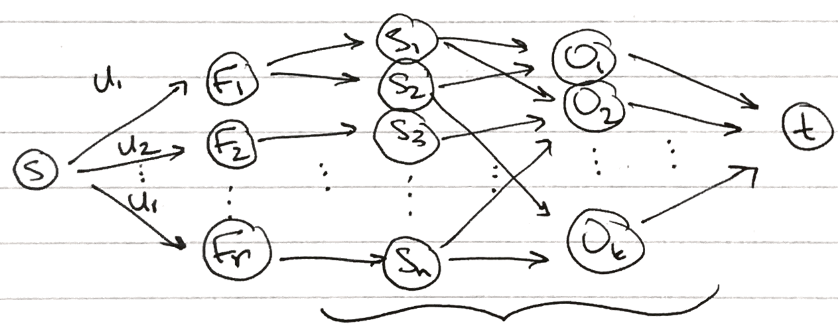
(1a) We will define the flow network $G' = (V', E')$ as follows. $s, t \in V$ will be in V'. For every vertex $v \in V - \{s, t\}$ there will be two vertices $V_{in}, V_{out} \in V'$. The edge $e(v) = (V_{in}, V_{out}) \in E'$ will have capacity $c'(v)$. Now, for every edge $e = (u, v) \in E$, where $u \neq s$ and $v \neq t$, the edge $(u_{out}, V_{in}) = e' \in E'$. If $u = s$, then the edge $e' = (s, V_{in}) \in E'$. If $v = t$, then the edge $e' = (u_{out}, t) \in E'$. The edge e', will have the capacity $c(e)$ where $e \in E$.



We then run Ford-Fulkerson on G' and get a max flow f. The $f(e')$ where $e' \in E'$ corresponds to to the $f(e)$ where $e \in E$. ($f(e') = f(e)$). Since $c(e') = c(e)$ $f(e) < c(e)$. Also, we know that the capacity of vertex V in G will not be exceeded since $f(e(v)) < c'(v)$.

The construction of the graph G' is done in $O(|V| + |E|)$. We also know that the runtime of Ford-Fulkerson is $O(c|E|)$ where c is the value of the max flow. Therefore the entire algorithm's runtime is bounded by $O(c|E|)$.

(1b) We will define the flow network $G=(V,E)$ as follows. Let $s,t \in V$, where s is the source and t is the sink. Let $O_1,\ldots,O_k \in V$, $F_1,\ldots,F_r \in V$, and $S_1,\ldots,S_n \in V$. For each $i \in [1,n]$, there exist an edge between S_i and their corresponding field of study F_j with capacity 1. Also there exist an edge between S_i and every organization O_k they are a part of. Each of these edges has a capicity of 1. There exists and edge (s,F_ℓ) for each $\ell \in [1,r]$ with capacity u_ℓ. Finally there exists an edge (O_γ,t) for all $\gamma \in [1,k]$ with capacity 1. For example,



These edges have a capacity of 1.

We then run ford-fulkerson on G and get a max flow f. If the value of $f(O_i,t) \neq 0$ for all $i \in [1,k]$, then a valid comittee can be formed. Since flow must be conserved, there must be exactly one incoming edge ~~for~~ to each organization O_i, therefore there are exactly k students in the comittee. Similarly, for each Student S_j and field F_ℓ, flow must be conserved. It follows from this that since $f(s,F_\ell) < u_\ell$ for all $\ell \in [1,r]$, there are not more than u_ℓ students on the comittee from each field F_ℓ. In conclusion all the requirements to form a comittee are met.

If there exists an edge $e = (O_i, t)$ s.t. $f(e) = 0$ then no valid comittee can be formed (or max flow less than k). There are two scenarios:

① There are less than k students.
② a comittee cannot be formed without exceeding the quota for a particular ~~case~~ field of study

In case ① it is clear that max flow will be less than k. Every student will be assigned to an Organization, therefore they may be organization with no flow through them

~~In case ② we can see that there will exist an edge e in G s.t. $f(e)$ = $f(s, F_e) = u_e$.~~

In case ②, for each field $f(s, F_e) = u_e$. This means that if flow was pushed through another student and organization then one of the capacities of ~~[illegible]~~ F_e will be exceeded.

The $|V| = n + r + k$ and $|E| = n + nk + r + k$. Therefore the runtime to construct the flow network G takes $O(|V| + |E|)$. To run Ford Fulkerson the cost is $O(C|E|)$ where $C = K$, Therefore the runtime of the entire algorithm is bounded by $O(K|E|)$.

(2a) <u>Lemma</u> If we augment a simple path P in the residual graph G', then the $v(f) < v(f')$ returned after augmenting.

<u>Pf.</u> Suppose e is the first edge in path P. Since P is simple we know that e must be a forward edge, therefore we add the bottleneck (P, f) to $f(e)$. Since all capacities are positive $f(e) < f(e) + \text{bottleneck}(P, f)$. It follows then that $v(f) < v(f) + \text{bottleneck}(P, f) = v(f')$.

∎

From this Lemma we can see that if there does not exist a path from s to t in G', then f must be a maximum flow since nothing can be augmented. Otherwise the flow f' after augmentation is greater than f. Since the max flow can only increase by k, the number of augmentations required is bounded by constant k.

<u>Algorithm</u>.

$G_f \longleftarrow$ residual Graph. with flow f and capacity function c'.
While there exists a simple path P in G_f
 $f \longleftarrow \text{augment}(f, c', P)$
 Update G_f.
return f.

The runtime of this is $O(k|E|)$ as explained above.

(2b) Algorithm. ($e^* = (u, v)$)

If $f(e^*) < c''(e^*)$
    return f

Let $\varepsilon = f(e^*) - c''(e^*) + 1$

Find simple path P_1 from t to v.
Find simple path P_2 from u to s.

Subtract ε from each f along the path P_1, e^*, P_2.

Let $G_f \longleftarrow$ residual graph with new f' and c''

while there exists a simple Path P in G_f
    $f' \longleftarrow$ augment (f', c'', P)
    update G_f.

return f'

Similar to (2a) we know that either the max flow remains the same or decreases by K. Therefore the max flow of the new network given c'' will be between $[f-k, f]$. From the lemma, it is clear then that the while loop will iterate atmost k times.

Since the length of path P_1, e^*, P_2 is at most $|V|-1$ and the cost of finding the path is $O(|V|+|E|)$ (BFS) The run time of the algorithm is dominated by the while loop. This bounded by $O(k|E|)$.