Flow Sample Problem

1. Where in the world is Carmen Sandiego?

Carmen Sandiego is currently in an $n \times n$ grid. From any square she can move to the four adjacent squares (up, down, left, right). Her goal is to get to any square on any "edge" of the grid, from which she can escape. You wish to prevent her escape; you will place ACME agents on various locations on the grid. Each grid location u is labeled with a weight $w(u)$ which denotes the number of agents needed to block Carmen from that location. If that many agents are placed on that location, Carmen cannot pass through. You have k agents to place; determine whether you can catch Carmen (and if so how to place the agents).

(a) Describe a graph you can run a max-flow algorithm on. Be sure to mention edge directions and capacities. **Solution:**

> Let $G = (V, E)$ be the original grid graph and v be the location of Carmen. Create a dummy source s and a dummy sink t. Add an edge from s to v, and an edge from every "edge" vertex to t. For each edge $e = (u, v) \in E$, replace it with both directed edges, i.e., (u, v) and (v, u). Set the capacities of all edges discussed so far to ∞.
>
> Now, for each vertex u of G, subdivide it into two vertices u_{in} and u_{out}. All edges entering u now enter u_{in}; all edges leaving u enter u_{out}. Add an edge $(u_{\text{in}}, u_{\text{out}})$ of weight $w(u)$.

(b) Describe how you'll tell whether an assignment is possible or not. If an assignment is possible, how do you read it from the result of the maximum flow algorithm? **Solution:**

> Run a max-flow algorithm with s as source and t as sink. Find the value of the maximum flow. If the value of the max flow is more than k, then there is no cut of value k so we cannot prevent Carmen's escape. This is because an allocation of k agents means we can reduce the weight over a finite cut by at most k. So a minimum cut of more than k means there is no possible allocation that reduces the value over the cut to zero. On the other hand, if the max flow is at most k, then we find a min-cut (using the standard process); this cut must use only finite weight edges (as k is finite). Thus the only cut edges must be of the form $(u_{\text{in}}, u_{\text{out}})$ for vertices u. All such vertices u should get $w(u)$ agents assigned to them.

(c) Briefly justify correctness. We aren't expecting a formal proof here, but you should have a sentence or two for each of the restrictions given in the problem. **Solution:**

> Any set of vertices that separate v from the border of the grid in the original graph will separate s from t (and vice versa) by construction. Because we subdivided each vertex (and made edges infinite capacity), the only finite cuts in the modified graph correspond to the weights on the vertices of the original graph. Thus any finite cut in the modified graph corresponds to a set of vertices to remove that separate v from the edges in the original graph.
>
> By the correctness of the max-flow algorithm, and the max-flow-min-cut theorem, we find the minimum cut, which is the minimum way to separate Carmen from her escape.

(d) Describe the runtime of the algorithm. Briefly justify why the running time is the value that you state. **Solution:**

> $\mathcal{O}(n^2 k)$. We have an upper bound of $2n^2 + n^2$ edges, corresponding to the grid edges plus the added vertex edges. We can also assume that for each vertex u in our graph, $w(u) \leq k$ or else we just return False. Hence the max flow in this case is at most k so using Ford-Fulkerson we get the runtime above.