MA421 Advanced Algorithms Lent 2019

Exercises 6

1. There may be several different minimum cuts in a graph. Using the analysis of Karger's randomised min-cut algorithm, argue that there can be at most $n(n-1)/2$ distinct minimum cuts.

2. Consider running the Karger's contraction algorithm (for min-cut problem) on graph G with n vertices until the number of vertices is reduced to t to obtain graph G' and, then, using a cubic-time deterministic algorithm to find a minimum cut in the contracted graph G'. First show that any specific min-cut survives in the graph G' with probability at least $\binom{t}{2}/\binom{n}{2}$. Now obtain an algorithm by repeating this process as many times as necessary to ensure a probability of success at least 0.99 to obtain an algorithm with running time $O(n^{8/3})$.

3. To improve the probability of success, Karger's min-cut algorithm can be repeated multiple times.

   (a) Consider running the algorithm twice. Determine the total number of edge contractions and give a (tight) lower bound on the probability of finding a minimum cut in the graph.

   (b) Now consider the following variation. Starting with a graph with n vertices, first contract the graph down to k vertices using contractions as in Karger's algorithm. Make ℓ copies of this graph with k vertices, and now run the randomised algorithm on these ℓ copies to the completion independently. At the end, output the minimum cut found in any of these ℓ repetitions. Determine the total number of edge contractions and give a lower bound on the probability of finding a minimum cut for this algorithm as a function of n, k, and ℓ.

   (c) Set the values of k and ℓ in terms of n, so that you achieve a better probability of success for the variation in (b), while keeping the total number of edge contractions in the two cases (a) and (b) roughly the same. Give an explanation as to why the approach in (b) achieves a better result.

4. Consider the following recursive improvement to Karger's randomised minimum-cut algorithm. This new algorithm proceeds in phases. In each phase, the algorithm first makes two copies of the current graph and works independently on each copy. The algorithm contracts randomly and independently chosen edges in each graph until the number of vertices in each copy decreases by a factor of $5/4$: for example, from n to $4n/5$. (You may assume that $4n/5$ is an integer.) Then, the algorithm proceeds to the next phase for each resulting graph. The algorithm runs, recursively and independently, on the two graphs obtained in this way, stopping recursion when $n \leq 3$ and outputting a minimum cut in the resulting graph. At the end, the algorithm returns the minimum of all cuts found by the recursion.

   You can visualise a complete binary tree representing the execution of this algorithm. The root corresponds to the input graph. Each arc in the binary tree represents one phase of the edge contractions on one of duplicates of the output graph from the previous phase. Analyse this algorithm by giving a lower bound on the probability of finding a minimum cut and a an upper bound on the number of edge contractions. Set the parameters to ensure success with probability at least $1 - \frac{1}{n}$.

[Hint: Let $P(n)$ denote the probability that the recursive algorithm described above succeeds on a graph with n vertices. Try to show that $P(n) \geq 1/\log_2 n$ using induction.]

5. Generalising the on the notion of a cut, we define a r-way cut in a graph as a set of edges whose removal breaks the graph into r or more components. Explain how Karger's min-cut algorithm can be used to find a minimum r-way cut and bound the probability that it succeeds in one iteration.

Solutions to these exercises are to be handed in at the lecture on **26 February 2019**.

