Exercise sheet 8

Exercises for the exercise session on 8th January 2025

Problem 25. Let $c > 0$ be a constant and set $p := \frac{c}{n^{2/3}}$. Use Janson's inequalities to determine a function $q\colon \mathbb{R}_{>0} \to (0,1)$ so that

$$\mathbb{P}[G(n,p) \text{ contains no clique of size } 4] \overset{n\to\infty}{\longrightarrow} q(c).$$

Problem 26. Let $c \geq 2$ be a constant and let $p = \left(\frac{c \ln n}{n^2}\right)^{\frac{1}{3}}$. Show that in $G(n,p)$ with high probability there is a path of length three between any pair of vertices.

(Hint : Use Janson's inequality to show that the probability that a fixed pair of vertices is not joined by such a path is $o\left(n^{-2}\right)$.)

Problem 27. Given an integer $k = k(n)$ with $1 \leq k \leq n$, consider the following randomised algorithm to determine the k-th smallest element of an arbitrary, but fixed array $(a[1], \ldots, a[n])$ of distinct numbers.

(i) Like in Randomised QuickSort, pick a pivot element $a[q]$ uniformly at random and sort all elements smaller than $a[q]$ to appear before $a[q]$ and all elements larger than $a[q]$ to appear after $a[q]$. Denote the resulting array by $(b[1], \ldots, b[n])$ and suppose that $a[q] = b[j]$.

(ii) If $j = k$, return $a[q] = b[j]$.

(iii) If $j < k$, go to step (i) for the array $(b[j+1], \ldots, b[n])$.

(iv) If $j > k$, go to step (i) for the array $(b[1], \ldots, b[j-1])$.

Denote by D_n the number of comparisons that this algorithm performs (in the same sense that comparisons were considered for Randomised QuickSort in the lecture). Prove that $\mathbb{E}[D_n] = \Theta(n)$.

Problem 28. Consider the following randomised algorithm which produces a *cut* in an n-vertex (multi-)graph G:
We set $G_n = G$ and given G_i we form G_{i-1} by choosing an edge of G uniformly at random and contracting that edge (that is, identify its two endpoints). We keep any parallel edges formed, and we note that every edge in G_{i-1} comes from a unique edge in G_i. We continue until G_2, where there are two vertices remaining. Let F be the set of edges of G associated to the edges of G_2.

(i) Show that F is a cut in G, that is, the removal of F splits G into multiple components;

(ii) Show that the probability that F is the smallest cut in G is at least $\frac{2}{n^2}$;

(iii) Devise a randomised algorithm which find the smallest cut in a graph whp.

(*Optional*) : Can this algorithm be implemented in polynomial time?