Institut für Informatik
Prof. Dr. Heiko Röglin
Dr. Melanie Schmidt

Randomized Algorithms and
Probabilistic Analysis
Summer 2016

Problem Set 7

Problem 1

Show that for all $n \geq k > 0$, it holds that

$$\left(\frac{n}{k}\right)^k \leq \binom{n}{k} \leq \left(\frac{n \cdot e}{k}\right)^k.$$

Hint: Recall Stirling's formula from the lecture. Additionally, it is helpful to write the binomial coefficient as

$$\binom{n}{k} = \frac{n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot (n-k+1)}{k \cdot (k-1) \cdot (k-2) \cdot \ldots \cdot 3 \cdot 2 \cdot 1}.$$

Problem 2

We consider 3-SAT and the situation that the input formula is satisfiable, let a^* be an arbitrary satisfying assignment. Assume that n is a multiple of two.

Based on the lower bound on q_i from the lecture, prove the easier to obtain bound $\Theta(\sqrt{2}^n \cdot \text{poly}(n, 1/\delta))$ on the number of necessary iterations to find a satisfying assignment with probability $1 - \delta$. Proceed by completing the following steps:

1. Let a be an assignment chosen uniformly at random. Let r be the probability that a agrees with a^* in at least $\frac{n}{2}$ variables. Show that $r \geq 1/2$.
2. Give a lower bound for the probability that in an iteration, the algorithm draws an a which agrees with a^* in at least $\frac{n}{2}$ variables and then reaches vertex n.
3. Based on 2., give a bound on the necessary number of iterations to find a satisfying assignment with probability $1 - \delta$.

Problem 3

Another easier to show upper bound for the number of iterations is of $\Theta((1.5)^n \cdot \text{poly}(n, 1/\delta))$. To show this bound, one uses a simpler lower bound for q_i: For each, $i \in \{0, \ldots, n\}$ the probability to reach n from $n - i$ is at least $(1/3)^i$ since the walk could go to n right away. Thus, the success probability of one iteration is at least

$$\frac{1}{2^n} \sum_{i=0}^{n} \binom{n}{i} \cdot \frac{1}{3^i} \cdot 1^{n-i} = \left(\frac{1}{2}\right)^n \cdot \left(1 + \frac{1}{3}\right)^n = \left(\frac{4}{2 \cdot 3}\right)^n = \left(\frac{2}{3}\right)^n,$$

yielding the above running time bound. Extend this result to a corresponding result for k-SAT. Is the resulting bound better than iterating through all possible assignments?