ShanghaiTech University

CS240 Algorithm Design and Analysis

Spring 2024

Problem Set 5

Due: 11:59pm, June 14, 2024

1. Submit your solutions to the course Gradescope.
2. If you want to submit a handwritten version, scan it clearly.
3. You are required to follow ShanghaiTech's academic honesty policies. You are allowed to discuss problems with other students, but you must write up your solutions by yourselves. You are not allowed to copy materials from other students or from online or published resources. Violating academic honesty can result in serious penalties.

Problem 1:

Suppose you have $2n$ balls and 2 bins. For each ball, you throw it randomly into one of the bins. Let X_1 and X_2 denote the number of balls in the two bins after this process. Prove that for any $\varepsilon > 0$, there is a constant $c > 0$ such that the probability $\Pr[X_1 - X_2 > c\sqrt{n}] \leq \varepsilon$.

Problem 2:

Suppose that for a certain decision problem, we have an algorithm which computes the correct answer with a probability at least 2/3 on any instance. We wish to reduce the error probability by running the algorithm n times on the same input, using independent randomness between trials, and taking the most common result as the final answer. Using Chernoff bounds, provide an upper bound on the probability that this modified algorithm produces an incorrect result.

Problem 3:

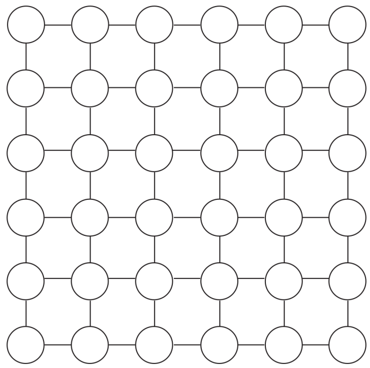
Suppose you have n coins, where the i'th coin has a size $c_i > 0$, and many piggy banks, each with a uniform capacity V, such that $V \geq \max(c_i)$. You want to place all the coins into the minimum number of piggy banks. To do this you use the following greedy strategy. Start with one active piggy bank. Then, sequentially go through the coins, attempting to place each coin into any active piggy bank where it fits. If a coin does not fit into any active piggy bank, take a new piggy bank and make it active. The algorithm is shown below. Prove this algorithm is a 2-approximation, i.e. it uses at most two times the minimum number of piggy banks needed for all the coins.

Algorithm 1 Piggy Bank Coin Packing

Input: Sizes of coins $c_1, c_2, \ldots, c_n$; size of piggy bank V
Output: Number of piggy banks used
Initialize $b \leftarrow 1$ ▷ current number of active piggy banks
Initialize $P_1, P_2, \ldots \leftarrow 0$ ▷ space used in each piggy bank
for $i = 1$ to n **do**
    if $\exists j \leq b$ such that $P_j + c_i \leq V$ **then**
        Choose any j with $P_j + c_i \leq V$
        $P_j \leftarrow P_j + c_i$ ▷ put c_i in j-th active piggy bank
    else
        $b \leftarrow b + 1$
        $P_b \leftarrow c_i$ ▷ open a new piggy bank and put c_i in it
    end if
end for

Problem 4:

Consider an $n \times n$ grid graph G, as shown below.



Each node v in G has a weight $w(v) > 0$. You want to choose an independent set of nodes with maximum total weight. That is, you want to choose a set of nodes S with maximum total weight such that for any $v \in S$, none of v's neighbors are in S. To do this, consider the following greedy algorithm. Let V be the set of all nodes in G. Choose the node in V with the largest weight (breaking ties arbitrarily), add it to the independent set, then remove the node and all its neighbors from V. Repeat this process until V is empty. Let S be the output of this algorithm. Solve the following problems.

1. Let T be any independent set in G. Show that for each node $v \in T$, either $v \in S$, or there is a neighbor v' of v with $v' \in S$ and $w(v) \leq w(v')$.

2. Show that the greedy algorithm is a 4-approximation.