ShanghaiTech University

CS240 Algorithm Design and Analysis

Spring 2020

Problem Set 4

Due: 23:59, June 5, 2020

1. Submit your solutions to Gradescope (www.gradescope.com).
2. In "Account Settings" of Gradescope, set your FULL NAME to your Chinese name and enter your STUDENT ID correctly.
3. If you want to submit a handwritten version, scan it clearly. Camscanner is recommended.
4. When submitting your homework, match each of your solution to the corresponding problem number.

Problem 1:

Given an undirected graph $G = (V, E)$, consider the following randomized algorithm for finding a min-cut in G, which improves the probability of returning a min-cut compared to the basic min-cut algorithm presented in class. The improved algorithm works by performing two independent contractions of G until it has $\lceil 1 + |V|/\sqrt{2} \rceil$ remaining vertices, producing graphs G_1 and G_2, then recursively computing the min-cuts of G_1 and G_2 and returning the smaller of the two cuts.

Give a lower bound on the probability that $cut2$ returns a correct min-cut.

Hint: Consider the probability that for a fixed min-cut C of G, there is no edge $e \in C$ that is contracted by $f(G, t)$.

```
Function f (G = (V, E), t)
    while |V| > t do
        choose e ∈ E randomly;
        contract edge e;
    end while
    return G ;
end
Function cut2 (G = (V, E))
    if |V| < 6 then
        return the min cut of G
    end if
    t ← ⌈1 + |V|/√2⌉;
    G1 ← f(G, t) ;
    G2 ← f(G, t) ;
    return smaller of cut2(G1) and cut2(G2) ;
end
```

Problem 2:

Again consider an undirected graph $G = (V, E)$

1. Give an upper bound on the maximum number of different min-cuts in G.

2. Give an efficient algorithm to find all min-cuts of G with probability, and argue that your algorithm is correct.

Hint: There is an elegant and simple solution which makes use of the basic min-cut algorithm presented in class.

Problem 3:

Recall that the 3-SAT problem asks whether it is possible to satisfy all the clauses in a 3-CNF. Since 3-SAT is NP-complete, we will try to solve an easier problem, namely satisfying as many clauses in a 3-CNF as possible. Given a 3-CNF ϕ, give an algorithm to find a truth assignment satisfying at least $7/8$ of the maximum number of satisfiable clauses in ϕ. Moreover, argue that there always exists an assignment satisfying at least $7/8$ of the number of clauses in ϕ.

Problem 4:

Suppose we have a device that generates a sequence of independent fair random coin-flips, but what we want is a six-sided die that generates the values $1, 2, 3, 4, 5, 6$ with *equal* probability. Give an algorithm that does so using $\frac{11}{3}$ coin-flips on average.

Problem 5:

In class, we saw that the expected time for randomized Quicksort to sort n numbers is $O(n \log n)$. Using Chernoff bounds, prove a high probability bound on this running time. That is, prove that for any $c > 0$, the probability randomized Quicksort finishes in $O(n \log n)$ time is $\geq 1 - \frac{1}{n^c}$.

Hint: In each recursive call to Quicksort, consider the event that we get a "good split", i.e. that the smaller partition contains at least one quarter of all the elements.