How many different minimum cuts are there in a tree with n nodes (ie. n−1 edges) ?

n2

n−1

n

2n−2

Let "output" denote the cut output by Karger's min cut algorithm on a given connected graph with n vertices, and let p=1n2. Which of the following statements are true? (For hints on this question, you might want to watch the short optional video on "Counting Minimum Cuts".)

There exists a graph G with n nodes and a min cut (A,B) of G such that

Pr[out=(A,B)]≤p.

For every graph G with n nodes and every min cut (A,B),

Pr[out=(A,B)]≤p.

For every graph G with n nodes and every min cut (A,B) of G,

Pr[out=(A,B)]≥p.

For every graph G with n nodes, there exists a min cut (A,B) such that

Pr[out=(A,B)]≤p.

For every graph G with n nodes, there exists a min cut (A,B) of G such that

Pr[out=(A,B)]≥p.

Let .5<α<1 be some constant. Suppose you are looking for the median element in an array using RANDOMIZED SELECT (as explained in lectures). What is the probability that after the first iteration the size of the subarray in which the element you are looking for lies is ≤α times the size of the original array?

2\*α - 1

1−α2

α−12

1−α

Let 0<α<1 be a constant, independent of n. Consider an execution of RSelect in which you always manage to throw out at least a 1−α fraction of the remaining elements before you recurse. What is the maximum number of recursive calls you'll make before terminating?

−log⁡(n)log⁡(1−α)

−log⁡(n)log⁡(α)

−log⁡(n)α

−log⁡(n)log⁡(12+α)

The minimum s-t cut problem is the following. The input is an undirected graph, and two distinct vertices of the graph are labelled "s" and "t". The goal is to compute the minimum cut (i.e., fewest number of crossing edges) that satisfies the property that s and t are on different sides of the cut.

Suppose someone gives you a subroutine for this s-t minimum cut problem via an API. Your job is to solve the original minimum cut problem (the one discussed in the lectures), when all you can do is invoke the given min s-t cut subroutine. (That is, the goal is to reduce the min cut problem to the min s-t cut problem.)

Now suppose you are given an instance of the minimum cut problem -- that is, you are given an undirected graph (with no specially labelled vertices) and need to compute the minimum cut. What is the minimum number of times that you need to call the given min s-t cut subroutine to guarantee that you'll find a min cut of the given graph?

n−1

2n

n

n

The following problems are for those of you looking to challenge yourself beyond the required problem sets and programming questions. Most of these have been given in Stanford's CS161 course, Design and Analysis of Algorithms, at some point. They are completely optional and will not be graded. While they vary in level, many are pretty challenging, and we strongly encourage you to discuss ideas and approaches with your fellow students on the "Theory Problems" discussion form.

1. Prove that the worst-case expected running time of every randomized comparison-based sorting algorithm is Ω(nlogn). (Here the worst-case is over inputs, and the expectation is over the random coin flips made by the algorithm.)
2. Suppose we modify the deterministic linear-time selection algorithm by grouping the elements into groups of 7, rather than groups of 5. (Use the "median-of-medians" as the pivot, as before.) Does the algorithm still run in O(n) time? What if we use groups of 3?
3. Given an array of n distinct (but unsorted) elements x1,x2,…,xn with positive weights w1,w2,…,wn such that ∑i=1nwi=W, a *weighted median* is an element xk for which the total weight of all elements with value less than xk (i.e., ∑xi<xkwi) is at most W/2, and also the total weight of elements with value larger than xk (i.e., ∑xi>xkwi) is at most W/2. Observe that there are at most two weighted medians. Show how to compute all weighted medians in O(n) worst-case time.
4. We showed in an optional video lecture that every undirected graph has only polynomially (in the number n of vertices) different minimum cuts. Is this also true for directed graphs? Prove it or give a counterexample.
5. For a parameter α≥1, an *α-minimum cut* is one for which the number of crossing edges is at most α times that of a minimum cut. How many α-minimum cuts can an undirected graph have, as a function of α and the number n of vertices? Prove the best upper bound that you can.