Problem 1 (Karger-Stein Algorithm). We’ve talked about Karger-Stein algorithm for MIN-CUT in class:

**Algorithm 1** Fast-Min-Cut(G)

```plaintext
if |V| ≤ 8 then
    Return Min-Cut(G)
else
    Let t = ⌈|V|/√2⌉.
    Let G₁ = Contract(G, t); G₂ = Contract(G, t);
    Return min{Fast-Min-Cut(G₁), Fast-Min-Cut(G₂)}.
end if
```

Here Min-Cut(G) is an algorithm for solving MIN CUT on small graphs; Contract(G, t) is similar to Karger’s algorithm and will randomly contract edges in G until there are only t vertices left.

In this problem we would like to show the success probability of the algorithm is Ω(1/log n) for a graph of n vertices.

(a) Let P(n) be the probability of success for the algorithm on n vertices, express P(n) recursively using P(n/√2). (Here for simplicity, you can assume n/√2 is an integer. Also, you can assume that the probability that the min cut of G₁ is equal to min cut of G is exactly 1/2.)

(b) Let f(t) = P(√2^t), use induction to prove that f(t) ≥ 4/t for t ≥ 4.

Problem 2 (Distinct Min Cuts). (a) Prove that any graph G with n vertices can have at most \(^{n \choose 2}\) distinct min cuts. (Two cuts are different if they contain different set of edges.)

(Hint: Look at the success probability of Karger’s min cut algorithm.)

(b) For any n, construct a graph that has exactly \(^n \choose 2\) distinct min cuts.