# Graded Exercise 9

Duong Le

## Problem 1

We have, applying the Master theorem:

$$c_{crit} = \log_{\sqrt{2}} 2 = 2$$

Also, we have that $O(n^2) = O(n^{c_{crit}} \log^0 n)$

$\Rightarrow T(n) = \Theta(n^{c_{crit}} \log n) = O(n^2 \log n)$

## Problem 2

We have:

$$P_1 = 1 - \frac{1}{2} \cdot 1^2 = \frac{1}{2} \leq p_0$$

Also:

$$\begin{aligned} & P_{d-1}^2 \geq 0 \\ \Rightarrow \quad & P_{d-1} \geq P_{d-1} - \frac{1}{2}P_{d-1}^2 = P_d \end{aligned}$$

From the two observations above, we get that:

$$P_d \leq P_{d-1} \leq 1$$

Also, since $P_{d-1} \leq 1 \rightarrow \frac{1}{2}P_{d-1}^2 \leq P_{d-1}$. Thus, $P_d = P_{d-1} - \frac{1}{2}P_{d-1}^2 \geq 0$

$\Rightarrow P_d \in [0, 1]$

### Base case

We have:

$$\frac{1}{0+1} = 1 \leq P_0$$

$\Rightarrow$ The base case is correct.

### Induction hypothesis

Assume that the hypothesis is correct for $d-1$, we need to prove that the hypothesis is also hold for $d$.

### Prove

We know that:

$$\begin{aligned} P_{d-1} &\geq \frac{1}{d} \\ P_d &= P_{d-1} - \frac{1}{2}P_{d-1}^2 \end{aligned}$$

We also know that the function $f(x) = x - \frac{1}{2}x^2$ is increasing in [0, 1], and we have shown that $P_{d-1} \in [0, 1]$, also it is clear that $\frac{1}{d} \in [0, 1]$. Thus, we have:

$$P_d \geq \frac{1}{d} - \frac{1}{2} \cdot \frac{1}{d^2}$$

Now, consider the following:

$$\begin{aligned} & 2d^2 \geq d^2 + d = d(d+1) \\ \Leftrightarrow \quad & \frac{1}{2d^2} \leq \frac{1}{d(d+1)} \\ \Leftrightarrow \quad & -\frac{1}{2d^2} \geq -\frac{1}{d(d+1)} \end{aligned}$$

Combine the two above results, we get:

$$P_d \geq \frac{1}{d} - \frac{1}{d(d+1)} = \frac{1}{d+1}$$

$\Longrightarrow$ Thus, the proof is complete.

# Problem 3

## b.

For each call of FASTMINCUT, the algorithm will make two recursive call on the graph of size $n/\sqrt{2}$, along with one call to the CONTRACT algorithm.

Also, in the first iteration, the runtime of the CONTRACT algorithm is $O((n-n/\sqrt{2})^2) = O(n^2)$. Thus, we can write the recurrence of $T(n)$ as:

$$T(n) = 2T\left(\frac{n}{\sqrt{2}}\right) + O(n^2)$$

And in problem 1, I have shown that the results of $T(n)$ is $O(n^2 \log n)$.

$\Rightarrow$ The runtime of the algorithm is $O(n^2 \log n)$