

Q1.A solution

According to the text, $f(i)$ is the depth of the first i characters of the string s . And $f(j)$ is the depth of the first j characters of the string s .

As $f(i) = f(j)$, $d(S_{\leq i}) = d(S_{\leq j})$, which means that

$$\#a(S_{\leq i}) - \#b(S_{\leq i}) = \#a(S_{\leq j}) - \#b(S_{\leq j})$$

By simplifying this equation, we get $\#a(S_{\leq j}) - \#a(S_{\leq i}) - (\#b(S_{\leq j}) - \#b(S_{\leq i})) = 0$.

It can be seen that $\#a(S_{\leq j}) - \#a(S_{\leq i})$ is the number of a in the string constructed by index $i+1$ to j in string s , $\#b(S_{\leq j}) - \#b(S_{\leq i})$ is the number of b in the string constructed by index $i+1$ to j in string s .

So, $s_{i+1}s_{i+2}\dots s_j$ is a weakly balanced string.

Q1.B solution

Define the set of strings that meet any of the three conditions in the problem as $P \subseteq \Sigma^*$, and define the set of all balanced strings as $F \subseteq \Sigma^*$. We are going to prove $P \subseteq F$ first.

The (i) case is $s = \epsilon$. It is obvious that $d(s)=0$ and $\#_a(s) \geq \#_b(s)$ since $\#_a(s) = \#_b(s) = 0$.

So, (i) is proved. Empty string is balanced.

In case (ii), as $x, y \in F$ and they are non-empty, by definition, we have $d(xy) = d(x) + d(y)$. As $d(x) = d(y) = 0$, $d(xy) = 0$. So $s = xy$ is also weakly balanced. Because x is a balanced string, any prefix substring p_1 of the first part of $s = xy$ satisfies $\#_a(p_1) \geq \#_b(p_1)$. Any prefix substring t of y satisfy $\#_a(t) \geq \#_b(t)$. $s = xy$ means adding the same number of a and b before y , then any prefix substring p_2 of the second part of $s = xy$ still satisfy $\#_a(p_2) \geq \#_b(p_2)$.

So, (ii) is proved.

In case (iii), assume $s_1 = a$, $s_2 = b$. By definition, $d(s_1) = 1$, $d(s_2) = -1$. Since we also have $d(xy) = d(x) + d(y)$, we can get $d(axb) = d(s_1) + d(x) + d(s_2) = 1 + d(x) - 1$. As x is a balanced string, $d(x) = 0$. So, $d(axb) = 0$, $s = axb$ is also weakly balanced. As x is a balanced string, any prefix substring p of x satisfies $\#_a(p) \geq \#_b(p)$. Add one a means that $\#_a(t) \geq \#_b(t)$ still right for any prefix substring t of string ax . As for the total string $s = axb$, $\#_a(s) = \#_b(s)$.

So, (iii) is proved. So, $P \subseteq F$ has been proved.

Now, we are going to prove that $F \subseteq P$.

Suppose there is a string $s \in F$ and the length of s is n , which means that s is a balanced string. We first classify according to whether the string is empty.

case (1): s is an empty string. in that case, $s \in P$.

case (2): s is a non-empty string. According to the definition of balanced string, we can know that " s must start with a ", because any prefix substring p of s satisfies

$\#_a(p) \geq \#_b(p)$, we can consider the prefix substring with only one letter. At the same time, "s must end with b", because for the (n-1) letter prefix substring, it satisfies $\#_a(p) \geq \#_b(p)$, and $\#_a(s) = \#_b(s)$. As s is balanced with $\#_a(s) = \#_b(s)$, "the string between the first 'a' and the last 'b' is a weakly balanced string".

So balanced string s can be expressed as aqb with weakly balanced string q. if q is empty, it is obvious that $s = aqb \in P$. So we then assume that q is a non-empty string. On the premise of case 2, we next classify s according to whether the string q is balanced.

case(2.1): q is a balanced string. In this case, $s = aqb \in P$ as it satisfies the requirements of (iii).

case(2.2): q is a weakly balanced string and q is not a balanced string.

- q is not a balanced string $\implies \exists p$, a prefix substring of q, we have that $\#_a(p) < \#_b(p)$. (1)

- aqb is a balanced string \implies for any prefix substring p of aq , we have that $\#_a(p) \geq \#_b(p)$. (2)

From (1) and (2), we can conclude that:

- For any prefix substring h of string q, it must satisfying $\#_a(h) \geq \#_b(h) - 1$. (3)
- q has at least one prefix substring m, which satisfies $\#_a(m) + 1 = \#_b(m)$. (4)
- For any prefix substring t of string m, it must satisfying $\#_a(t) + 1 \geq \#_b(t)$. (5)

Using (4), We just find and cut the prefix substring m, and split the q into string m and string n. Now, if we can prove that am and nb are both non-empty balanced strings like what (ii) says, then we can prove that in case(2.2), $q \in P$.

First of all, it is obvious that because of $\#_a(m) + 1 = \#_b(m)$, $\#_a(n) - 1 = \#_b(n)$, am and nb are both weakly balanced strings.

For am , using (5), when you add one "a" before m, we can conclude that "for any prefix substring t of string am , they must satisfying $\#_a(t) \geq \#_b(t)$ ", so am is balanced string.

For nb , suppose $\#_a(m) = x$, then $\#_b(m) = (x + 1)$, using (3), when you delete m before $q = mn$, we can conclude that "for any prefix substring h of string q, there is a corresponding prefix substring c of string n, which must satisfying $\#_a(c) = \#_a(h) - x \geq \#_b(h) - x - 1 = \#_b(c)$ ".

Thus, am and nb are both non-empty balanced strings. In case (2.2), $q \in P$. Because in case(1), case(2.1), case(2.2), we can derive that $q \in P$, we prove that $F \subseteq P$. Thus, $F = P$ and we have proved that if a string is balanced, it is either belongs to (i)(ii)(iii) in the question.

Q1.C solution

Let $f(i) = d(s_{\leq i})$. If $f(i) = f(j)$ with $i < j$, then $s_i s_{i+1} \dots s_j$ is a weakly balanced string, which is proved in 1.A. It suggests that everytime the value of $f(i)$ repeats, we can cut in the corresponding position to get weakly balanced substring. If there are m - 1 repetitive values of $f(i)$ for a certain value of i, then we can cut m - 1 times, and get m substrings.

Then we use contradiction to prove question C. Suppose there exists a balanced string $s_c \in \{a, b\}^*$ of length n such that it doesn't satisfy condition (i) and (ii). For formally,

- The maximum depth of s_c is $< \sqrt{n}$ and

• s_c can't be broken into m non-empty substrings $s_c = t_1|t_2|...|t_m$ such that t_2, t_3, \dots, t_{m-1} are weakly balanced strings and $m \geq \sqrt{n} - 1$

To let s_c fail to satisfy condition (ii), we try to find minimum possible value of m . This situation happens when there are minimum repetitive values of $f(i)$ for a given i . By definition of $f(i)$ we can get the following formula for $f(i)$ with $i \geq 1$:

$$f(i) = \begin{cases} f(i-1) + 1, & s_i \text{ is a,} \\ f(i-1) - 1, & s_i \text{ is b} \end{cases}$$

As we can see in the formula, alternating as and bs increase the number of repetitive values of $f(i)$. So the situation happens in the case of least number of alternating as and bs, i.e. all as first and then b, like $aaaa...bbbb..aaaa..bbbb$. As maximum depth of s_c is $< \sqrt{n}$, the minimum number of $aaaa..bbbb..$ is $\frac{n}{2\sqrt{n}} - 1 = \frac{\sqrt{n}}{2} - 1$. The minimum number of repetitive values of $f(i)$ is $(\frac{\sqrt{n}}{2} - 1) * 2 = \sqrt{n} - 2$ and minimum possible value of m is $\sqrt{n} - 1$, which means such s_c doesn't exist. So any balanced string must satisfy condition either (i) or (ii).

Q2 solution

We use induction to solve this problem.

Base case: Let $n = 2$. There are only two people in the world and if there's only one tribe with 2 people, no lambs are sacrificed. If there are two tribes each with 1 person, then one tribe will lose and one lamb is sacrificed. In both cases. The number of lambs sacrificed is smaller than $n \log_2 n = 2$

Let $n = 3$, Then then number of lambs sacrificed can be 0 or 1, both of which is smaller than $3 \log_2 3$.

Inductive hypothesis: For any $n > 1$, at most $n \log_2 n$ lambs got sacrificed.

Induction step: Let W be a world of n people, Assume inductive hypothesis holds for all worlds with number of people greater than 2 and less than n .

Then in the world with $n-1$ people, we have number of lambs sacrificed $\leq (n-1) \log_2 (n-1)$. By adding one more person, this world will have n people. If the tribe this new-added person is in wins all the fight, then the number of lambs sacrificed doesn't change. If the the tribe this new-added person is in loses, there will be one more lamb sacrificed. In this case, we have

$$\begin{aligned} \text{Number of lambs sacrificed in world } W &\leq (n-1) \log_2 (n-1) + 1 \\ &= n \log_2 (n-1) + 1 - \log_2 (n-1) \\ &\leq n \log_2 (n-1) \\ &< n \log_2 (n) \end{aligned}$$

In both cases, the number of lambs sacrificed in world W is less than $n \log_2 (n)$

So during this process, at most $n \log_2 n$ lambs got sacrificed.

Q3.A solution

Intuition: Assume that $\exists t \in Z$ and $\exists C \in R$, for any $n \in Z$ satisfying $n \leq t$ and $n \geq 4$, we can get $T(n) \leq Cn \log n$. Now, we are attempting to prove that for $n = t + 1$, $T(n) \leq Cn \log n$.

- since $n \geq 4$, $\lfloor n/4 \rfloor < n \leq t$, $\lfloor 3n/4 \rfloor < n \leq t$,

$$\begin{aligned}
T(n) &= 2n + T(\lfloor n/4 \rfloor) + T(\lfloor 3n/4 \rfloor) \\
&\leq 2n + \frac{Cn}{4} \log \frac{n}{4} + \frac{3Cn}{4} \log \frac{3n}{4} \\
&= 2n + \frac{Cn}{4} (\log n - \log 4) + \frac{3Cn}{4} (\log n + \log \frac{3}{4}) \\
&= 2n + Cn \log n + \frac{3Cn \log 3}{4} - Cn \log 4
\end{aligned}$$

- In order to let $T(n) \leq Cn \log n$,

$$\begin{aligned}
2n + \frac{3Cn \log 3}{4} - Cn \log 4 &< 0 \\
2 + \frac{3C \log 3}{4} - C \log 4 &< 0 \\
C &> \frac{2}{\log 4 - \frac{3}{4} \log 3} \approx 8.189
\end{aligned}$$

Formal answer: Let $C = 9$ and set a constant b satisfying $b \geq 1$. We assume that $T(n) \leq 9n \log n + b$.

• **base case:** For $n = 1$ and $n = 2$, it is obvious that $T(1) = 1 \leq b$ and $T(2) = 1 \leq 18 \log 2 + b$.

• **Induction hypothesis:** Let $k > 0$ be an arbitrary integer, assume that $T(n) \leq 9n \log n + b$ for any integer n satisfying $n \leq k$.

• **Induction Step:** Now, if we prove that $T(n) \leq 9n \log n + b$ holds for $n = k + 1$, then we can conclude that $T(n) \leq 9n \log n + b$.

case 1 ($k \leq 8$): $T(n) = 1$ for n satisfying $n = k + 1 \leq 9$. Let $f(n) = 9n \log n + b$. Because $f'(n) = 9 + 9 \log n > 0$ for $n \geq 1$, $f(n)$ is an increasing function. We can prove that $T(n) \leq f(1) \leq f(n) = 9n \log n + b$.

case 2 ($k \geq 9$): $T(n) = 2n + T(\lfloor n/4 \rfloor) + T(\lfloor 3n/4 \rfloor)$ for n satisfying $n = k + 1 \geq 10$. since $n = k + 1 \geq 10$, we could know that $\lfloor n/4 \rfloor < n \leq k$ and $\lfloor 3n/4 \rfloor < n \leq k$. Thus,

$$\begin{aligned}
T(n) &= 2n + T(\lfloor n/4 \rfloor) + T(\lfloor 3n/4 \rfloor) \\
&\leq 2n + \frac{9n}{4} \log \frac{n}{4} + \frac{27n}{4} \log \frac{3n}{4} + 2b \\
&= 2n + \frac{9n}{4} (\log n - \log 4) + \frac{27n}{4} (\log n + \log \frac{3}{4}) + 2b \\
&= 2n + 9n \log n + \frac{27n \log 3}{4} - 9n \log 4 + 2b \\
&= -0.1980n + 9n \log n + 2b
\end{aligned}$$

In order to let $T(n) \leq 9n \log n + b$, we can get $b \leq 0.1980n$. Since b is any constant that satisfying $b \geq 1$ and $n = k + 1 \geq 10$, we can simply let $b = 1$.

$$T(n) = -0.1980n + 9n \log n + 2 \leq 9n \log n + 0.02 \leq 9n \log n + 1 = 9n \log n + b$$

Now, we can conclude that $T(n) \leq 9n \log n + 1$ for all natural number n . According to the definition, we prove that $T(n) = O(n \log n)$ for natural number n . Finally, we

consider the case where n is a negative integer or 0. Obviously, $T(n) = 1$, so $T(n) = O(1)$ for negative integer or 0. Overall, $T(n) = O(n \log n)$ for all integer n .

Q3.B solution Intuition: Assume that $\exists t \in \mathbb{Z}$ and $\exists C \in \mathbb{R}$, for any $n \in \mathbb{Z}$ satisfying $n \leq t$ and $n \geq 1$, we can get $T(n) \leq Cn$. Now, we are attempting to prove that for $n = t + 1$, $T(n) \leq Cn$.

- since $n \geq 1$, $\lfloor n/4 \rfloor \leq n \leq t$, $\lfloor n/6 \rfloor \leq n \leq t$, $\lfloor n/7 \rfloor \leq n \leq t$

$$\begin{aligned} T(n) &= n + T(\lfloor n/2 \rfloor) + T(\lfloor n/6 \rfloor) + T(\lfloor n/7 \rfloor) \\ &\leq n + \frac{Cn}{2} + \frac{Cn}{6} + \frac{Cn}{7} \end{aligned}$$

- In order to let $T(n) \leq Cn$,

$$\begin{aligned} n + \frac{Cn}{2} + \frac{Cn}{6} + \frac{Cn}{7} &\leq Cn \\ n &\leq \frac{4Cn}{21} \\ C &\geq \frac{21}{4}, (n \geq 1) \end{aligned}$$

Formal answer: Let $C = 10$. We assume that $T(n) \leq f(n) = 10n$.

- **base case:** For $n = 1$ and $n = 2$, it is obvious that $T(1) = 1 \leq 10 = f(1)$ and $T(2) = 1 \leq 20 = f(2)$.

- **Induction hypothesis:** Let $k > 0$ be an arbitrary integer, assume that $T(n) \leq 10n$ for any integer n satisfying $n \leq k$.

- **Induction Step:** Now, if we prove that $T(n) \leq 10n$ holds for $n = k + 1$, then we can conclude that $T(n) \leq 10n$.

case 1 ($k \leq 22$): $T(n) = 1$ for n satisfying $n = k + 1 \leq 23$. Let $f(n) = 10n$. Because $f'(n) = 10 > 0$ for $n \geq 1$, $f(n)$ is an increasing function. We can prove that $T(n) \leq f(1) \leq f(n) = 10n$.

case 2 ($k \geq 23$): $T(n) = n + T(\lfloor n/4 \rfloor) + T(\lfloor n/6 \rfloor) + T(\lfloor n/7 \rfloor)$ for n satisfying $n = k + 1 \geq 24$. since $n = k + 1 \geq 24$, we could know that $\lfloor n/4 \rfloor \leq n \leq k$, $\lfloor n/6 \rfloor \leq n \leq k$ and $\lfloor n/7 \rfloor \leq n \leq k$. Thus,

$$\begin{aligned} T(n) &= n + T(\lfloor n/4 \rfloor) + T(\lfloor n/6 \rfloor) + T(\lfloor n/7 \rfloor) \\ &\leq n + \frac{10n}{4} + \frac{10n}{6} + \frac{10n}{7} \\ &= \frac{277}{42}n \\ &\approx 6.5952n \leq 10n \end{aligned}$$

Now, we can conclude that $T(n) \leq 10n$ for all integer n . According to the definition, we prove that $T(n) = O(n)$. Finally, we consider the case where n is a negative integer or 0. Obviously, $T(n) = 1$, so $T(n) = O(1)$ for negative integer or 0. Overall, $T(n) = O(n)$ for all integer n .