ADA homework1
b03902089 林良翰

problem 1.(1).(a)

claim：exist positive integers n_0, c_1, c_2, s.t. for all $n \geq n_0$, $F(n) < c_1n^2 - c_2n$

proof：suppose $F(n) \leq c_1n^2 - c_2n$ for n = m-1, m-2…

when n = m：

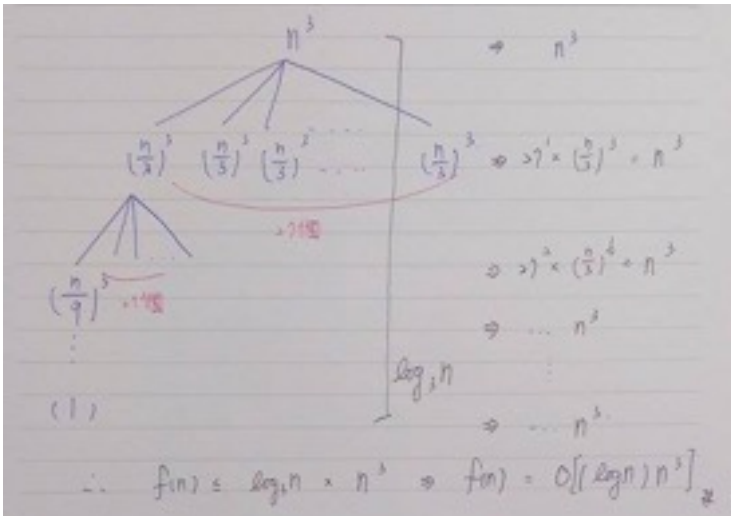
$$
\begin{aligned}
F(n) &= 16F(n/4) + 514n \\
&\leq 16(c_1(n/4)^2 - c_2(n/4)) + 514n \\
&= c_1n^2 - 4c_2n + 514n \\
&= c_1n^2 - c_2n - (3c_2 - 514)n
\end{aligned}
$$

thus we can know $c_2 \geq 514/3$

besides, we have to fit the initial conditions：

so we can let $c_2 = 514/3$ and $c_1 = 517/3$, $n_0 = 1 \Rightarrow F(n) = \mathbf{O}(n^2)$

problem 1.(1).(b)



problem 1.(2)

// define $\log_2(x) = \lg(x)$

$F(n) = eF(n/2) : \mathbf{\Theta}(n^{\log e})$

$F(n) = eF(n/2) = e^2F(n/4) = \ldots = e^{\log n}F(1) = n^{\log e}$

$F(n) = F(n-1) + n^e : \mathbf{\Theta}(n^{e+1})$

$F(n) = n^{1/2}F(n^{1/2}) + n : \mathbf{\Theta}((\log n)^2\log(\log n))$

let $k = \log n$:

$2^k = n$, $n^{1/2} = 2^{k/2}$

$F(2^k) = 2^{k/2}F(2^{k/2}) + 2^k$

let $G(x) = F(2^{k/2})$

$G(x) = x/2G(x/2) + x$

$G(x) = x^2\log x$

so, $F(n) = (\log n)^2\log(\log n)$

$F(n) = F(n-1) + 1/n : \mathbf{\Theta}(1)$

$F(n) = F(n-1) + F(n-2) : \mathbf{\Theta}(2^n)$

suppose there exist integer c and n_0, for $n < n_0$, such that $F(n) < c2^n$:

while $n = n_0$

$F(n) = F(n-1) + F(n-2)$

$\leq c2^{n-1} + c2^{n-2}$

$\leq c(2^{n-1}+2^{n-1})$

$= c2^n$

$n^{1/\lg(n)}$: $\Theta(1)$

by limit calculator : https://www.symbolab.com/

limit $(n \to \infty)$ $n^{1/\lg(n)} = 2$

sorted results :

$\Theta(1)$: f(n-1) + 1/n, 2147483647, 2^{10000}, F(n) = F(n-1) + 1/n, $n^{1/\lg(n)}$

$\Theta(\log(\log(n)))$:, $\lg(\ln(n))$

$\Theta(\log(n))$: $\sum(1/i)$ for 1 to n

$\Theta((\log n)^2\log(\log(n)))$: $F(n) = \sqrt{n}F(\sqrt{n}) + n$

$\Theta((\log(n))^{\log(n)})$: $(\lg(n))^{\ln(n)}$

$\Theta(n)$: $e^{\ln(n)}$, 10n/e

$\Theta(n^{\log\log n})$: $n^{\lg(\lg(n))}$,$n^{\lg(\ln(n))}$

$\Theta(n/\log(n))$: $n/\ln(n)$

$\Theta(n^{\ln(\sqrt{2})})$: $\sqrt{2}^{(\ln(n))}$

$\Theta(n\log(\log(n)))$: $en\ln(\lg(n))$

$\Theta(n\log(n))$: $n\lg(n)$, $n\ln(n)$, $\ln(n!)$

$\Theta(n^{\log(e)})$: F(n) = eF(n/2)

$\Theta(n^{3/2})$: $n^{3/2}$

$\Theta(n^3)$: $e^5n^3 - 10n^2 + e^{1000}$

$\Theta(n^{e+1})$: $F(n) = F(n-1) + n^e$

$\Theta(2^n)$: F(n) = F(n-1) + F(n-2)

$\Theta(n^n)$: n!

reference : graph by Google

problem 2.(1).(a)

```
function findMajority(*arr, head, end)
    if (head == end) return arr[head]
    else if (head+1 == end)
        if (arr[head] == arr[end]) return arr[head]
        else return -1
        end if
    else
        mid = (head+end)/2
        front = findMajority(arr, head, mid)
        behind = findMajority(arr, mid+1, end)
        if (front == behind != -1) return front
        else if (front == -1 & behind != -1) return behind
```

```
                else if (front != -1 & behind == -1) return front
                else return -1
                end if
        end if
end function
```

problem 2.(1).(b)

```
function findMajority(*arr, size)
        count = 0, result = arr[0]
        for i from 0 to size-1
                if (arr[i] == result) count++
                else count−−
                end if
                if (count == 0) result = arr[i]
                end if
        end for
        count = 0
        for i from 0 to size-1
                if (arr[i] == result) count++
                end if
        end for
        if (count > size/2) return result
        else return -1
        end if
end function
```

reference: http://www.geeksforgeeks.org/majority-element/

problem 2.(2).(a)

$T(n) = 2n + 3n + \ldots + kn \Longleftarrow$ first merge 1+1 arrays, next compare 1+2 arrays…

$= (2 + 3 + \ldots + k)\ n$

$= ((2+k)(k-1)/2)\ n$

$= (k^2+k-2)/2\ n$

$= \mathbf{O}(k^2 n)$

problem 2.(2).(b)

$arr_1, arr_2, arr_3, \ldots arr_k$

```
function arrayMerge(head, end)
        if (head == end) return arr_head
        else if (head+1 == end)
                return new_arr = merge of arr_head, arr_end
        else
                mid = (head+end)/2
                new_arr1 = arrayMerge(head, mid)
                new_arr2 = arrayMerge(mid+1, end)
```

return new_arr = merge of new_arr1, new_arr2
end if
end function

$T(n) = (2n)k/2 + (4n)k/4 + \ldots$
$= nk \times \log(k)$
$= \mathbf{O}(nk\ \mathit{log}k)$

problem 3.(1)

suppose AB, CD, Ef… have a minimum total segment length.
proof : if AB, CD, Ef… is a miserable-word match of AB intersect CD, than we can find a smaller total segment match with AC and BD ⇒contradict with suppose

⇒ if there is a minimum total segment length, then it is a good-word match

problem 3.(2)

step 1. find a point **x** arbitrarily, and set it as origin with x-axis and y-axis.
step 2. connect the segment L_1 between origin and point P_1, and compute the angle Θ_1 between y-axis and L_1
step 3. compute Θ_2 to Θ_N similarly
step 4. sort all Θs (**O**(N*log*N))
step 5. set a line L_x vertically through origin, rotate the line clockwise with angle Θ1, Θ2… until we find an angle such that the right side of the L_x have the same number of white points and black points (d = 0)

problem 3.(3)

according to problem 3.(3), we need **O**(N*log*N) for finding a line to divide the plain, so we need :

$\mathbf{O}(N^2 \mathit{log} N)$ to find all lines (N lines) for good-word match