

Trabalho 1 - 2018-2: Linear Selection

Professor: Molinaro

Rio de Janeiro, 27 de setembro 2018

Grupo:

Luis Claudio

Rodrigo Pumar

1. Código da Linear Selection

```
def linearSelection(lst, l, r, k):
    if (k > 0 and k <= r - l + 1):
        n = r-l+1 # Number of elements in lst[l..r]
        # Divide lst[] in groups of size 5, calculate
        # median of every group and store it in median[].
        median = [0]*int((n+4)/5) #// There will be floor((n+4)/5) groups;
        fat = 5
        i = 0
        while i < int(n/5):
            median[i] = selectMedian(lst[fat-5:fat])
            fat = fat + 5
            i = i + 1
        if i*5 < n: #//For last group with less than 5 elements
            median[i] = selectMedian(lst[fat-5:])
            i = i + 1
        if i == 1:
            medMed = median[i-1]
        else:
            medMed = linearSelection(median, 0, i-1, i/2)

        #// Partition the list around a random element and
        #get position of pivot element in sorted list
        lst, pos = partition(lst, l, r, medMed)
        #print('Part\t',pos)

        #// If position is same as k
        if (pos-l == k-1):
            return lst[pos]
        if (pos-l > k-1): #// If position is more, recur for left
            return linearSelection(lst, l, pos-1, k)

        #// Else recur for right sublist
        return linearSelection(lst, pos+1, r, k-pos+1-1)
    return -1
```

2. Complexidade

Essa seleção linear implementada, possui complexidade expressa pela expressão: $T(n) \leq \text{cst} \cdot n + T(n/5) + T(7n/10)$

Sendo $\text{cst} \cdot n = A1 + A2$

2.1.A1

Cada recursão calcula a mediana para cada grupo de 5, tendo custo de ordenar cada grupo $5\log(5)$ e selecionar o elemento da mediana. Faz se isso $n/5$ vezes.

$$A1 = 5\log(5) \cdot \text{cst} \cdot n/5 = \log(5) \cdot \text{cst} \cdot n$$

2.2.A2

A partição da lista, tem custo $\text{cst} \cdot n$, pois temos que percorrer toda a lista e separar de acordos com a medianas das medianas. A cada elemento da lista de n elementos, fazemos cst operações.

3. Execução e Entrada

A entrada é dada pelo arquivo file.txt.

O formato do arquivo deve ser números separados por virgulas, no formato exemplo abaixo:

5,10,20,50,30,6,9,4,3,7

Para execução, crie esse arquivo “file.txt” na mesma pasta em que se executa o arquivo incluído (.py).