

Bases de Probabilidade, Problema da Seleção

Espaço amostral: U é o conjunto de todos os resultados possíveis.

- Cada resultado r em U tem uma probabilidade $P_r(r) \geq 0$

- e $\sum_{r \in U} P_r(r) = 1$

Eventos: um evento é um subconjunto $S \subseteq U$ e $P_r(S) = \sum_{r \in S} P_r(r)$

Variáveis aleatórias: uma variável aleatória X é uma função $X: U \rightarrow \mathbb{R}$

Cenário 1: resultado da rolagem de 1 dado.

- $U = \{1, 2, 3, 4, 5, 6\}$ e $P_r(r) = 1/6 \forall r \in U$

Evento $\rightarrow$ Resultado ser par, i.e., $S = \{2, 4, 6\}$. $Pr(S) = \sum_{r \in S} P_r(r) = |S| \cdot 1/6 = 3/6 = 1/2$

Variável $\rightarrow$ X é 1 se o resultado é par.

Cenário 2: resultado da rolagem de 2 dados.

- $U = \{(1,1), (1,2), (1,3), \dots, (6,5), (6,6)\}$ e $P_r(r) = 1/36 \forall r \in U$

Evento $\rightarrow$ Soma dos resultados ser 7, i.e., $S = \{(1,6), (2,5), (3,4), (4,3), (5,2), (6,1)\}$

- $Pr(S) = \sum_{r \in S} P_r(r) = |S| \cdot 1/36 = 6/36 = 1/6$

Variável $\rightarrow$ Y é a soma dos resultados dos dados.

Esperança: a esperança de uma variável aleatória $X: U \rightarrow R$ é

- a média de seus valores ponderada pelas probabilidades dos mesmos,
 - i.e., $E[X] = \sum_{n \in U} X(n) P_U(n)$

Linearidade da esperança: sendo X e Y variáveis aleatórias definidas sobre U ,

- a linearidade da esperança diz que $E[X + Y] = E[X] + E[Y]$

- Demonstração:

$$E[X + Y] = \sum_{n \in U} (X(n) + Y(n)) P_U(n) = \sum_{n \in U} X(n) P_U(n) + \sum_{n \in U} Y(n) P_U(n) = E[X] + E[Y]$$

- Note que, a propriedade vale mesmo que as variáveis sejam dependentes.
- Quiz1: generalizar para a soma de n variáveis $X_1, X_2, X_3, \dots, X_n$.

Cenário 1: resultado da rolagem de 1 dado. $U = \{1, 2, 3, 4, 5, 6\}$ e

- $\Pr(r) = 1/6$ para r em U . X é 1 se o resultado é par.

$$E[X] = \sum_{n=1}^6 X(n) P_U(n) = \sum_{n=1}^6 X(n) \frac{1}{6} = \frac{1}{6} (X(1) + X(2) + X(3) + X(4) + X(5) + X(6)) = 3/6 = 1/2$$

Cenário 2: resultado da rolagem de 2 dados. $U = \{(1,2), (1,3), \dots, (6,5), (6,6)\}$ e

- $\Pr(r) = 1/36$ para r em U . Y é a soma dos resultados dos dois dados.

- Seja Y_1 e Y_2 o resultado de cada dado. Note que $Y = Y_1 + Y_2$

- Temos que $E[Y_1] = E[Y_2] = \sum_{n=1}^6 Y_2(n) P_U(n) = \sum_{n=1}^6 n \cdot \frac{1}{6} = \frac{1}{6} (1+2+3+4+5+6) = \frac{21}{6} = 3,5$

- Assim, $E[Y] =$

$$E[Y_1 + Y_2] = E[Y_1] + E[Y_2] = 3,5 + 3,5 = 7$$

usando a linearidade da esperança

Problema da seleção

No problema da seleção, dado um vetor com n elementos em qualquer ordem, queremos determinar o k -ésimo menor elemento.

- Este problema generaliza o problema de encontrar a mediana de um vetor.

Podemos resolver este problema reduzindo-o ao problema da ordenação, ou seja, ordenamos o vetor e devolvemos o elemento que terminou na posição k .

- Como conhecemos algoritmos de ordenação que levam tempo $O(n \log n)$, esta abordagem também tem essa complexidade de tempo.

Mas, será que conseguimos fazer melhor que isso?

- Afinal, ao menos intuitivamente, ordenar parece bem trabalhoso do que encontrar o elemento que ocupa uma certa posição na ordem.

Como motivação podemos considerar um caso particular deste problema,

- que é encontrar o mínimo ou o máximo. Sabemos fazer isso em tempo linear.

Será que conseguimos usar divisão e conquista

- para resolver o problema da seleção em tempo linear?

Nos inspirando no algoritmo quickSort, vamos usar o algoritmo particiona,

- que resolve o problema da separação em tempo linear,
- e a ideia de escolher um pivô aleatoriamente.

selecao(vetor v, int k):

se tam(v) == k: devolva v[k]

p = escolhePivo(v)

(vl, vr) = particiona(v, p)

m = tam(vl) + 1

se k = m: devolva v[m]

se k < m: devolva selecao(vl, k)

senão /* k > m */ devolva selecao(vr, k-m)

Note que, no algoritmo anterior, supomos que k está no intervalo [1, tam(v)].

Prova de corretude: Basta aplicar indução matemática no tamanho do subproblema. Fica como exercício.

Análise de eficiência do algoritmo recursivo para seleção

Qual a eficiência de pior caso deste algoritmo? Ou seja, qual o pior tipo de pivô que podemos escolher?

- R.: O menor elemento (ou o maior). Neste caso, a cada chamada recursiva eliminamos apenas um elemento.
- Qual a eficiência neste caso? R: $\Theta(n^2)$.
- Isso porque o particiona percorre todo o vetor em cada chamada. Assim, no total temos $n + n-1 + n-2 + \dots + 3 + 2 + 1 = n(n-1)/2$ iterações do particiona.

Mas este caso é bastante raro se escolhermos o pivô aleatoriamente.

Qual a eficiência no melhor caso deste algoritmo? I.e., qual o melhor tipo de pivô?

- R.: Se desconsideramos a sorte de escolhermos justamente o k-ésimo elemento como pivô, o melhor tipo de pivô é o que divide o vetor ao meio.
- Neste caso especial teríamos a seguinte recorrência: $T(n) = T(n/2) + O(n)$
 - A que função de tempo essa recorrência corresponde?

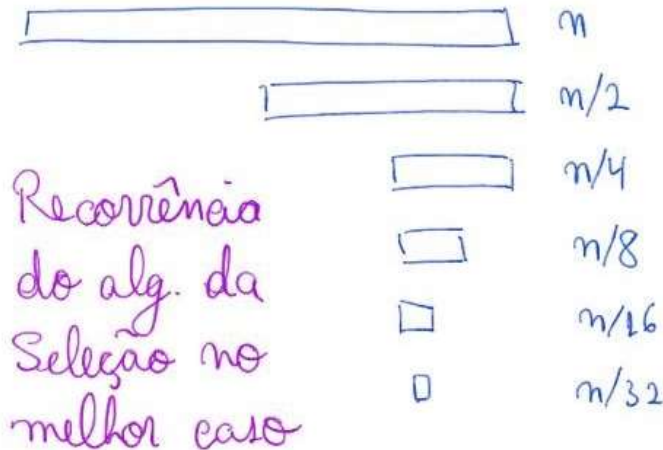
Lembrando do Teorema mestre: $T(n) = a T(n/b) + c n^d$

1. $a = b^d \rightarrow T(n) = O(n^d \log n)$
2. $a < b^d \rightarrow T(n) = O(n^d)$
3. $a > b^d \rightarrow T(n) = O(n^{\log_b a})$

Assim, para $T(n) = T(n/2) + O(n)$ temos $a = 1$, $b = 2$ e $d = 1$.

- Como $a = 1 < 2^1 = b^d$ caímos no caso 2 do Teorema Mestre.
- Portanto, $T(n) = O(n^1) = O(n)$, i.e., tempo linear no tamanho da entrada.

$$T(n) = T(n/2) + O(n)$$



Qual a eficiência esperada para este algoritmo?

- Para esta análise vamos usar o conceito de fase.
- Dizemos que o algoritmo selecao está na fase j se tam ,
 - o tamanho do vetor atual, está em $n \cdot (3/4)^{j+1} < \text{tam} \leq n \cdot (3/4)^j$.
 - Note que, no início o algoritmo está na fase 0.
- Seja X_j uma variável aleatória que indica
 - o número de chamadas recursivas realizadas pelo algoritmo na fase j .
- O trabalho não recursivo que o algoritmo selecao faz é $\leq c \cdot \text{tam}$,
 - já que a subrotina particiona leva tempo linear.
- Assim, podemos contabilizar o trabalho total do algoritmo selecao como
$$\begin{aligned} T(n) &\leq \sum_{j \in \text{fases}} X_j \cdot c \cdot \text{tamanho atual do vetor} \\ &\leq \sum_{j \in \text{fases}} X_j \cdot c \cdot n \cdot (3/4)^j \\ &= cn \sum_{j \in \text{fases}} X_j \cdot (3/4)^j \end{aligned}$$

Mas, quanto deve durar a fase j ?

- Definimos um bom pivô como aquele que gera uma partição 25%-75%,
 - ou seja, que quebra o vetor corrente em dois de forma que
 - nenhum é menor que 25% ou maior que 75%.
- Note que, metade dos elementos do vetor corrente são bons pivôs, ou seja,
 - a probabilidade de escolher um bom pivô aleatoriamente é $50\% = 1/2$.
- Note também que a fase sempre termina quando escolhemos um bom pivô.
 - Quiz2: Por que?

Agora, considere a variável geométrica Y com probabilidade de sucesso igual a $1/2$.

- Variável geométrica é aquela que contabiliza o número de lances de uma "moeda" com probabilidade de sucesso p até obter o primeiro sucesso.
- Podemos calcular a esperança de Y como segue

$$E[Y] = 1 + 1/2 \cdot E[Y] \rightarrow E[Y] - 1/2 \cdot E[Y] = 1 \rightarrow 1/2 E[Y] = 1 \rightarrow E[Y] = 2$$

- Por curiosidade, no geral a esperança de uma variável geométrica
 - com probabilidade de sucesso p é $1/p$.

Como uma fase sempre termina quando um pivô bom é escolhido

- e a probabilidade de um pivô bom ser escolhido é $1/2$, temos que
 - a esperança de Y é um limitante superior para a esperança de X_j ,
 - i.e., para todo j temos $E[X_j] \leq E[Y] = 2$.
- Por que limitante superior? R: Porque uma fase também pode terminar com uma divisão ruim. Quiz3: Como?

Voltando à nossa equação principal do tempo de execução

$$E[T(n)] \leq E[c \cdot n \sum_{j \in \text{fases}} X_j (3/4)^j]$$

(usando linearidade da esperança)

$$\leq c \cdot n \sum_{j=0+\infty} E[X_j] (3/4)^j$$

$$\leq c \cdot n \sum_{j=0+\infty} E[Y] (3/4)^j$$

$$\leq c \cdot n \sum_{j=0+\infty} 2 (3/4)^j$$

$$\leq 2 \cdot c \cdot n \sum_{j=0+\infty} (3/4)^j$$

$$\leq 2 \cdot c \cdot n \frac{1}{1-(3/4)}$$

$$= 2 \cdot c \cdot n \frac{4}{4-3}$$

$$= 8 \cdot c \cdot n = O(n)$$

sendo que a última desigualdade deriva da fórmula para

- a soma dos termos de uma PG de razão menor que 1.

The image shows handwritten mathematical work. On the left, the formula for the sum of a geometric series is written: $qS_{\infty} = \sum_{j=0}^{\infty} q^{j+1}$. Below it, the formula for the sum of a geometric series is written: $S_{\infty} = \sum_{j=0}^{\infty} q^j$. A large curly bracket connects these two formulas to the equation $(1-q)S_{\infty} = 1 \Rightarrow S_{\infty} = 1/(1-q)$. To the right of the first formula, the text "Soma dos termos de uma Progressão" is written in purple. To the right of the second formula, the text "Geométrica de razão menor que 1" is written in purple.

$$qS_{\infty} = \sum_{j=0}^{\infty} q^{j+1}$$
$$S_{\infty} = \sum_{j=0}^{\infty} q^j$$
$$(1-q)S_{\infty} = 1 \Rightarrow S_{\infty} = 1/(1-q)$$

Soma dos termos de uma Progressão

Geométrica de razão menor que 1