**1 O Bolo de Apostas**

questao1.cpp

precisa de input dos casos de teste

**2 Merge/Insertion Sort**

questao2.cpp

gera os próprios casos de test

Resultado de questao2.cpp na minha Máquina:

Nas seguintes quantidades de elementos no array:

CPU time in ms for insertionSort:

CPU time in ms for mergeSort:

10 20 30 40 100 200 300 400 1000 2000 3000 4000

0.001 0.001 0.002 0.003 0.015 0.057 0.138 0.221 1.358 5.423 12.347 21.977

0.002 0.002 0.003 0.003 0.008 0.015 0.024 0.032 0.087 0.182 0.291 0.399

Conclusão:

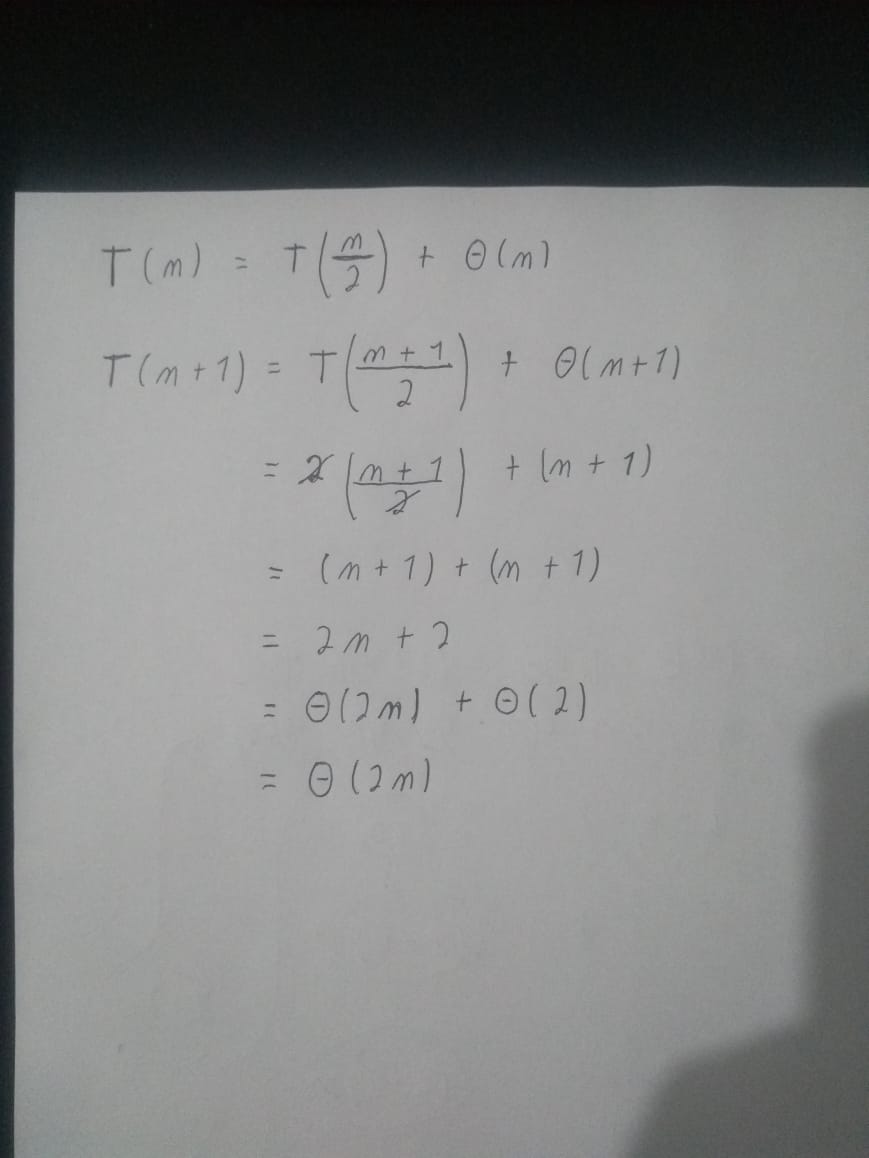
* 10 ~ 40 elementos o insertionSort se mantém igual ou melhor ao mergeSort
* 100 ~ 400 elementos o mergeSort se mantém melhor que o insertionSort
* 1000 ~ 4000 elementos o mergeSort se mantém muito melhor que o mergeSort

n0 é aproximadamente de 40 elementos

**3 Complexidade do Quick-Select**

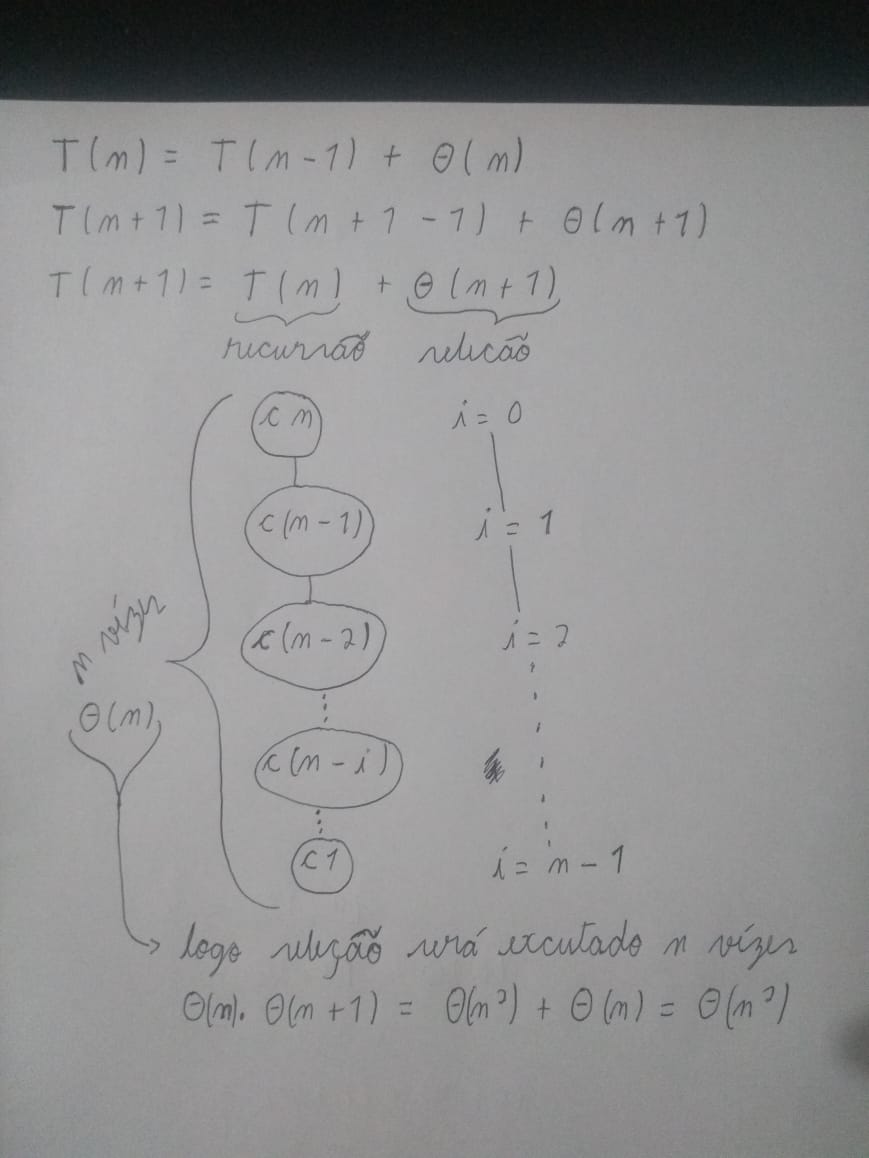
**Mostre por indução que T(n) = O(2n)**

questao3\_parte1.jpeg

****

**Mostre usando árvore de recursão que T(n) = O(n^2)**

questao3\_parte2.jpeg



**4 Quick-Select**

questao4.cpp

gera os próprios casos de test

**5 Mediana de Dois Arrays**

questao5.cpp

precisa de input dos casos de teste