 UNIVERSIDADE DE COIMBRA FACULDADE DE CIÊNCIAS E TECNOLOGIA *Departamento de Engenharia Informática*	**Trabalho nº 1A de Algoritmos e Estruturas de Dados** **2017-2018 – 2º Semestre** LEI **Data de Entrega Tarefa 1A:** 2 de Março de 2018 23:00, submissão no Mooshak.
<u>Nota Importante</u>: A fraude denota uma grave falta de ética e constitui um comportamento não admissível num estudante do ensino superior e futuro profissional licenciado. Qualquer tentativa de fraude pode levar a anulação da componente prática tanto do facilitador como do prevaricador.	

Esta ficha é constituída por três tarefas:

1A: Pretende-se que o aluno consolide conhecimentos sobre a importância da complexidade O-Grande de um algoritmo na viabilidade ou não da respetiva implementação. Na análise de complexidade vamo-nos concentrar no fator tempo**.**

1B e 1C: Implementação e aplicação de estruturas de dados (a disponibilizar proximamente).

Conceitos: Algoritmos

Um algoritmo deve ter as seguintes propriedades:
(www.tutorialspoint.com/data_structures_algorithms/algorithms_basics.htm)

- **não ser ambíguo:** conjunto de instruções claro e sem ambiguidades;
- **entradas (dados):** deve ter ZERO ou mais dados de entrada bem definidos;
- **saídas (resultados):** deve ter UM ou mais resultados bem definidos;
- **finitude:** deve terminar ao fim de um número finito de passos;
- **viabilidade:** deve ser viável com os recursos disponíveis;
- **independência:** deve definir uma execução passo a passo, independente da linguagem usada.

Conceitos: Análise de Complexidade

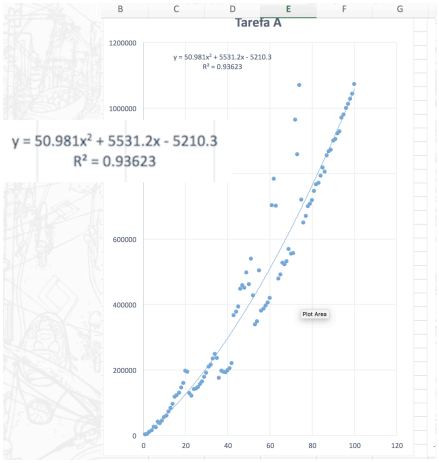
Análise *a priori*: é uma análise teórica da complexidade do algoritmo, efetuada antes de este ser implementado e corrido… vamos falar sobre isto mais tarde.

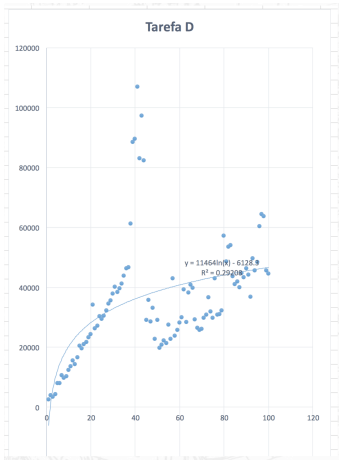
Análise *a posteriori*: é uma análise empírica da complexidade do algoritmo, realizada sobre uma implementação do mesmo. A implementação é corrida e um conjunto de estatísticas são recolhidas.

O presente trabalho 1A tem como objetivo central fazer a análise a posteriori dos algoritmos a implementar.

Exemplo de uma análise a posteriori para a implementação de um algoritmo estudado nas sessões teóricas.

A	B	C	D
2966	27634	22244	2439
4319	4299	1972	2446
8949	6166	2140	3393
13255	9178	2640	4347
16833	14594	3213	8452
23732	17680	4943	8934
30356	17714	4150	7533
36447	20465	6096	9912
45170	24069	5128	9618
56300	29916	7030	12147
65112	31172	10108	13743
73251	36880	6772	12996
82780	40225	8342	14114
96433	44051	9206	16627
117062	48169	9569	17806
120369	52890	10361	20007
128716	56173	10829	20749
146861	61141	16189	35066
159141	68488	12429	23810
212484	70017	12428	24791





As figuras acima representam, da esquerda para a direita, uma tabela com valores, neste caso em mS, da corrida de um programa nas versões A a D para um conjunto de dados de dimensão crescente (indicados em abcissa nos gráficos central e à direita); a figura central representa os resultados e a função f(N) para a versão A. A figura à direita representa os resultados, e a função f(N) para a versão D.

Conclui-se neste caso que a versão A tem complexidade assimptótica, de acordo com a análise empírica realizada, de $O(N^2)$ e a versão D tem complexidade O(N logN).

Relatório

O relatório a realizar sobre esta parte do trabalho (com base no formulário disponibilizado) e que deve fazer parte integrante do relatório para o trabalho #1 deve incluir e ter em conta as seguintes recomendações:

- definir a escala (ex. linear, logarítmica) de valores para as dimensão dos dados de entrada;
- não esquecer de colocar no eixo das ordenadas a unidade de tempo utilizada
- incluir gráficos para a solução exaustiva e melhorada de dimensões que permitam fácil leitura, sem ocupar demasiado espaço
- reflectir e elaborar sobre possíveis razões para o aparecimento de outliers (ex. os que aparecem no gráfico à direita na figura)
- concluir sobre em que medida os resultados obtidos são ou não os esperados, justificando a conclusão.

É esperado que no final da realização do trabalho o aluno:

- tenha clara perceção do impacto da complexidade O-grande na viabilidade de um algoritmo;
- reconheça que quanto temos uma solução inviável em termos de complexidade temporal (e/ou espacial) precisamos de revisitar o desenho do algoritmo e estruturas de dados utilizadas.
- saiba distinguir entre análise *a priori* e *a posteriori* e tenha ganho experiência na preparação e obtenção de conclusões de uma análise *a posteriori*.

Problema :: Par de pontos mais próximos num plano.

Dados n pontos num espaço bidimensional, calcular a distância entre os pontos com a menor distância entre si.

Como entrada o programa (quer para a tarefa A quer para a B) recebe uma linha com o número de pontos no plano, exemplo:

16[\n]

seguido do número de linhas necessárias para ter todas as linhas com 10 pontos representados pelas suas coordenadas X e Y e os restantes na última linha. As coordenadas são representadas por valores do tipo int na gama [0 .. 9999999]. Todos os valores são separados por espaço, com excepção do último valor em cada linha, a que se segue \n, exemplo:

34 2 21 298765 3 4 45 54 54 12 23 37 3 8 9 12 32 34 76654 897[\n]
345 5 54 6 67 23 12 456 65 23 78 65[\n]

A menor distância calculada é um valor do tipo decimal e deve ser apresentada com 3 casas decimais.

Exemplos de entrada e saída do programa (tarefas A e B)

ENTRADA
12[\n]
1 0 2 2 2 0 0 0 2 0 1 2 2 2 0 2 0 1 1 1[\n]
1 0 2 0[\n]

SAÍDA
0.000[\n]

ENTRADA
10[\n]
92 29 17 86 33 85 95 17 79 4 36 96 41 32 73 16 13 82 8 86[\n]

SAÍDA
5.657[\n]

Tarefa :: Solução Exaustiva

Resolver o problema recorrendo a uma solução exaustiva (calcular a distância de todos os pontos a todos os restantes).

Tarefa :: Solução Melhorada

Resolver o problema recorrendo a uma solução melhorada.

Referências:

Introduction to Algorithms, 2nd Edition
Thomas H. Cormen, Charles Leiserson, Ronald Rivest, Clifford Stein
The MIT Press, 2001, 2005

Closest Pair Problem (PowerPoint)
Subhash Suri
Disponível na página de AED e em
http://www.cs.ucsb.edu/~suri/cs235/ClosestPair.pdf

Tarefa :: Relatório :: Formato

O relatório, deve fazer uso do *template disponibilizado também no infoestudante* e que compreende os seguintes pontos:

- Gráfico Excel incluindo regressão para a solução exaustiva.
- Gráfico Excel incluindo regressão para a solução melhorada.
- Conclusões sobre a complexidade O-grande de cada algoritmo e sucinta análise crítica do resultado.
- Reflexão crítica sobre possíveis outliers (caso estes apareçam na sua análise empírica).

Bom trabalho,

Carlos Lisboa Bento