

# Trabalho Prático 1

Raul H. C. Lopes

25 de Junho de 2008

## 1 Objetivo do trabalho

O presente trabalho representa uma continuação do exercício realizado no trabalho anterior. Nosso objetivo aqui consiste em estudar o uso de estruturas de indexação espacial e árvore de prioridade com alocação e desalocação explícita em *heap*.

## 2 O problema em foco

Neste trabalho estaremos focalizados exclusivamente na heurística *Farthest Insertion* do *Symmetric TSP (STSP)*. Dado um conjunto de pontos no plano Euclidiano de duas dimensões, objetiva-se encontrar o menor *tour* que passa por todos os pontos. Assuma que voce quer viajar por todos os pontos e voltar ao ponto de partida e que a distância entre dois pontos quaisquer é a distância Euclidiana. Mais informações sobre este problemas e a heurística em foco podem ser encontradasno capítulo 9 de[1], *Experimental Analysis of Heuristics for the STSP*, de autoria de Johnson and McGeoch.

## 3 Os exercícios

Você implementará em *C* a heurística *Farthest Insertion*, contendo obrigatoriamente:

- Uma estrutura de prioridade, que permita a realização de qualquer operação em  $O(\lg n)$ .
- Uma estrutura de indexação espacial para cálculo de vizinho mais próximo em tempo médio de  $O(\lg n)$ .

- Uma representação eficiente para o tour que permita inserção de novo ponto em  $O(\lg n)$ .

Seu objetivo consiste em:

- propor duas soluções que usem filas diferentes e representações do tour diferentes.
- definir conjecturas sobre eficiência dos seus algoritmos;
- implementar seus algoritmos;
- testar exaustivamente seus algoritmos;
- relatar sobre seus testes, comparando resultados obtidos com as conjecturas iniciais.

Em todo o trabalho você está restrito a usar os algoritmos aqui especificados e as estruturas apresentadas em sala de aula.

## 4 Entrada e Saída

Seu programa lerá os pontos para computação do *tour* de *standard input*. Um ponto por linha: coordenada  $x$ , seguida de espaço, seguida de coordenada  $y$ .

Seu programa produzirá em *standard output* os pontos do *tour*. Um ponto por linha: coordenada  $x$ , seguida de espaço, seguida de coordenada  $y$ .

Qualquer desvio desta especificação implicará na anulação do trabalho.

O trabalho deve ser submetido para

raulh.ufes at gmail dot com

Qualquer falha em compilar implicará na anulação do trabalho.

## 5 Referências úteis

Algumas referências úteis como fontes de algoritmos:

- Estruturas de dados espaciais: *Samet, H.*, “Foundations of Multidimensional and Metric Data Structures”.
- Árvores binárias e árvores B: *Knuth, D.E.*: “The Art of Computer Programming”, Vol I e Vol III.
- Self-adjusting structures: *Tarjan, R., Sleator, D.*, “Self-adjusting binary trees”.

## 6 Avaliação dos trabalhos

O trabalho deverá ser desenvolvido individualmente.

Os trabalhos poderão ser usados para substituir parcialmente as duas provas. A seguinte escala será usada:

- Conclusão eficiente de dois dos quatro exercícios do primeiro trabalho e segundo trabalho garante aprovação com 6.
- Conclusão completa dos dois trabalhos garante aprovação com 8.
- Implementação de exercício extra do segundo trabalho garante aprovação com?

## 7 Datas importantes

- Entrega de proposta de implementação do trabalho 1 e conclusão do trabalho 0: 30 de junho.
- Entrega da implementação do trabalho 1: 7 de julho.
- Entrega de implementação e relatórios de testes dos trabalhos: 10 de julho.

## Referências

- [1] Gregory Gutin and Abraham P. Punen. *Traveling Salesman Problem and Its Variations*. Kluwer Academic Publishers, 2002.