U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Trabalho Prático 1

Grupo 99:

- Martim Henriques, up202004421
- Rúben Viana, up202005108
- Tiago Barbosa, up202004926

Descrição do Problema

Uma empresa tecnológica pretende inovar, criando uma plataforma eletrónica de crowdsourcing para a entrega de mercadorias em zonas urbanas. A empresa tem o seu próprio armazém para guardar as encomendas e recorre a um conjunto de estafetas para realizar as entregas. A empresa pretende criar uma plataforma que lhe permita distribuir as encomendas pelos estafetas de forma a minimizar o número de carrinhas usadas ou maximizar o lucro total da empresa. Esta plataforma também deverá permitir minimizar o tempo médio de entrega das encomendas expresso.

Descrição da Solução

Cenário 1 - Formalização

Neste cenário pretende-se distribuir as encomendas pelas carrinhas de forma a minimizar o número de estafetas usados para a entrega de todos os pedidos ou do maior número de pedidos , num dia.

Sendo C o vetor de n carrinhas. Sendo E o vetor das m encomendas de cada carrinha.

Objetivo : Min $\sum_{k=1}^{n} Ck.u$ onde u é 1 se a carrinha for usada ou 0 se não.

Sujeito a :

$\sum_{k=1}^{m} C \rightarrow Ek.p \leq C.pm$ onde p representa o peso das encomendas e pm o peso máximo da carrinha e E o vetor das encomendas na carrinha.

$\sum_{k=1}^{m} C \rightarrow Ek.v \leq C.vm$ onde v representa o volume das encomendas e vm o volume máximo da carrinha e E o vetor das encomendas na carrinha.

Descrição da Solução

Cenário 1 – Descrição de algoritmos relevantes

Para resolver este cenário optamos por uma estratégia gananciosa tendo como inspiração o algoritmo First Fit Decreasing de Bin Packing. Assim ordenamos as carrinhas e as encomendas de acordo com o seu tamanho de forma decrescente.

```
bool Empresa::compararCarrinhasCenario1(Carrinha &lhs, Carrinha &rhs) {
    return lhs.getVolMax()*lhs.getPesoMax() > rhs.getVolMax()*rhs.getPesoMax();
}

bool Empresa::compararEncomendasCenario1(Encomenda &lhs, Encomenda &rhs) {
    return lhs.getVolume()*lhs.getPeso() > rhs.getVolume() * rhs.getPeso();
}
```

Depois fomos colocando as encomendas por ordem nas carrinhas verificando sempre se estava dentro dos limites do volume e peso.

Descrição da Solução

Cenário 1 – Análise de complexidade

Baseado no algoritmo First Fit Decreasing, depois de ordenar o vetor de carrinhas e encomendas, percorremos para cada encomenda o vetor de carrinhas até encontrar uma em que a encomenda possa ser colocada pelo que esta solução tem complexidade temporal O(n x m) onde n é o número de carrinhas e m o número de encomendas e complexidade espacial O(1).

```
for (Encomenda &encomenda : encomendasDoDia) {
    bool adicionada = false;
    for (Carrinha &carrinha : carrinhas){
        if(encomenda.getVolume()>carrinha.getVolMax()){ //encomenda não cabe mesmo estando carrinha vazia
            break;
        }
        if(carrinha.adicionarEncomenda(encomenda)){
            adicionada = true;          //encomenda adicionada a carrinha
            break;
        }
    }
    if(!adicionada){ //entra aqui se todas as carrinhas estiverem cheias e esta encomenda não couber
        encomendasNaoEntregues.push_back(encomenda);
    }
}
```

Descrição da Solução

Cenário 1 – Resultados da avaliação empírica

10 encomendas -> 3 111 575[ns]

50 encomendas -> 5 463 587[ns]

100 encomendas -> 5 852 909 [ns]

200 encomendas -> 8 522 150 [ns]

450 encomendas -> 10 009 134 [ns]

A avaliação empírica não está de acordo com a complexidade $O(n^2)$ no entanto isto pode se explicar visto que no caso médio o "for" interior nunca corre na sua totalidade visto que é terminado assim que encontra uma carrinha para colocar a encomenda pelo o que o tempo de execução não aumenta tanto como o esperado.

Descrição da Solução

Cenário 2 - Formalização

Neste cenário pretende-se distribuir as encomendas pelas carrinhas de forma a maximizar o lucro da empresa para a entrega de todos os pedidos ou do maior número de pedidos , num dia.

Sendo C o vetor de n carrinhas. Sendo E o vetor das m encomendas de cada carrinha.

Objetivo : Max $\sum_{k=1}^{n} Ck.u * ((\sum_{k=1}^{m} Ck \rightarrow Ek.r) - Ck.c)$ onde Ck.u é 1 se a carrinha for usada ou 0 se não e Ek.r é a recompensa de cada encomenda e Ck.c é o custo da carrinha.

Sujeito a :

$\sum_{k=1}^{m} C \rightarrow Ek.p \leq C.pm$ onde p representa o peso das encomendas e pm o peso máximo da carrinha e E o vetor das encomendas na carrinha.

$\sum_{k=1}^{m} C \rightarrow Ek.v \leq C.vm$ onde v representa o volume das encomendas e vm o volume máximo da carrinha e E o vetor das encomendas na carrinha.

Descrição da Solução

Cenário 2 – Descrição de algoritmos relevantes

Para resolver este cenário voltamos a optar por uma estratégia gananciosa como o algoritmo First Fit Decreasing de Bin Packing.

A primeira forma de ordenação de encomendas dá o lucro total máximo mas pode deixar bastantes encomendas por entregar. Na segunda opção apesar de o lucro poder ser um pouco mais baixo entregamos a maior parte das encomendas. Deixamos o utilizador escolher a abordagem.

Depois ,tal como no cenário 1, fomos colocando as encomendas por ordem nas carrinhas verificando sempre se estava dentro dos limites do volume e peso.

```
bool Empresa::compararCarrinhasCenario2(Carrinha &lhs, Carrinha &rhs) {
    return (lhs.getVolMax()*lhs.getPesoMax())/lhs.getCusto() > (rhs.getVolMax()*rhs.getPesoMax())/rhs.getCusto();
}

bool Empresa::compararEncomendasCenario2LucroMaximo(Encomenda &lhs, Encomenda &rhs) {
    return lhs.getRecompensa()/(lhs.getVolume()*lhs.getPeso()) > rhs.getRecompensa()/(rhs.getVolume() * rhs.getPeso());
}

bool Empresa::compararEncomendasCenario2LucroEquilibrado(Encomenda &lhs, Encomenda &rhs) {
    return lhs.getRecompensa()*lhs.getVolume()*lhs.getPeso() > rhs.getRecompensa()*rhs.getVolume() * rhs.getPeso();
}
```

Descrição da Solução

Cenário 2 – Análise de complexidade

Tal como no cenário 1 é usado a mesma abordagem gananciosa logo esta solução tem complexidade temporal O(n x m) onde n é o número de carrinhas e m o número de encomendas e complexidade espacial O(1).

```
for (Encomenda &encomenda : encomendasDoDia) {
    bool adicionada = false;
    for (Carrinha &carrinha : carrinhas){
        if(encomenda.getVolume()>carrinha.getVolMax()){ //encomenda não cabe mesmo estando carrinha vazia
            break;
        }
        if(carrinha.adicionarEncomenda(encomenda)){
            adicionada = true;          //encomenda adicionada a carrinha
            break;
        }
    }
    if(!adicionada){ //entra aqui se todas as carrinhas estiverem cheias e esta encomenda não couber
        encomendasNaoEntregues.push_back(encomenda);
    }
}
```

Descrição da Solução

Cenário 2 – Resultados da avaliação empírica

10 encomendas -> 2 854 062 [ns]

50 encomendas -> 4 109 946 [ns]

100 encomendas -> 4 937 306 [ns]

200 encomendas -> 7 885 947 [ns]

450 encomendas -> 9 863 822 [ns]

A avaliação empírica não está de acordo com a complexidade $O(n^2)$ no entanto isto pode se explicar visto que no caso médio o "for" interior nunca corre na sua totalidade visto que é terminado assim que encontra uma carrinha para colocar a encomenda pelo o que o tempo de execução não aumenta tanto como o esperado.

Descrição da Solução

Cenário 3 - Formalização

Neste cenário pretende-se entregar as encomendas expresso de forma a minimizar o tempo médio previsto das entregas expresso a serem realizadas num dia.

Sendo E o vetor de n encomendas expresso do dia.

Objetivo : Min $(\sum_{k=1}^{n} Ek.u * \mathrm{Ek.t} \,/\, \sum_{k=1}^{n} Ek.u)$ onde Ek.u é 1 se a encomenda for entregue ou 0 se não e Ek.t é o tempo de duração da entrega da encomenda.

Sujeito a :

$\sum_{k=1}^{n} Ek.t * Ek.u \leq 8\ hrs$ visto que as encomendas expresso têm que ser entregues entre as 9 e as 17 horas.

Descrição da Solução

Cenário 3 – Descrição de algoritmos relevantes

Baseamo-nos no algoritmo de job scheduling para minimizar o tempo médio de entrega da encomenda ordenando as encomendas de forma crescente da sua duração e entregamo-las de acordo com essa ordem verificando se chegamos ao final do dia e se sobrarem encomendas ficam para serem entregues no dia seguinte.

```
std::sort(encomendasDoDiaExpresso.begin(), encomendasDoDiaExpresso.end(),
          [](Encomenda e1, Encomenda e2){return e1.getDuracao() < e2.getDuracao();}); // ordenar encomendas por duracao decrescente

int intervaloTempoEntregas = 28800; // das 9:00 as 17:00
int index = 0;
while (intervaloTempoEntregas > 0 && index < encomendasDoDiaExpresso.size()){
    intervaloTempoEntregas -= encomendasDoDiaExpresso[index].getDuracao();
    encomendasExpressoEntregues.push_back(encomendasDoDiaExpresso[index]);
    index++;
}

for (int i = index; i < encomendasDoDiaExpresso.size(); ++i) {
    encomendasExpressoNaoEntregues.push_back(encomendasDoDiaExpresso[i]); //encomendas nao entregues
}
```

Descrição da Solução

Cenário 3 – Análise de complexidade

Neste caso a função sort é de maior complexidade que a parte de ir entregando as encomendas pelo o que a complexidade temporal é O(nlogn) onde n é o número de encomendas e a complexidade espacial O(1).

std::sort <algorithm>

default (1)	`template <class RandomAccessIterator>` `  void sort (RandomAccessIterator first, RandomAccessIterator last);`
custom (2)	`template <class RandomAccessIterator, class Compare>` `  void sort (RandomAccessIterator first, RandomAccessIterator last, Compare comp);`

Sort elements in range

Sorts the elements in the range `[first,last)` into ascending order.

The elements are compared using `operator<` for the first version, and *comp* for the second.

Equivalent elements are not guaranteed to keep their original relative order (see `stable_sort`).

Complexity

On average, linearithmic in the distance between *first* and *last*: Performs approximately $N*\log_2(N)$ (where *N* is this distance) comparisons of elements, and up to that many element swaps (or moves).

Descrição da Solução

Cenário 3 – Resultados da avaliação empírica

10 encomendas -> 2 014 772 [ns]

50 encomendas -> 5 025 354 [ns]

100 encomendas -> 5 281 210 [ns]

200 encomendas -> 6 808 535 [ns]

450 encomendas -> 8 033 480 [ns]

O tempo de execução está como esperado numa complexidade O(n*logn) visto que como é logarítmica não aumenta tanto para inputs maiores comparando a inputs mais pequenos , explicando assim o pouco aumento do tempo de execução.

Funcionalidade extra

Encomendas não entregues passam para o dia seguinte com prioridade

Decidimos implementar uma funcionalidade extra no cenário 3 que quando chegava às 17 horas do dia e ainda sobravam encomendas expresso para entregar estas passavam para o dia seguinte com prioridade em comparação com as encomendas já planeadas para esse dia.

```
if (!encomendasExpressoNaoEntregues.empty()){
    waitEnter(); // pressionar tecla para continuar
    cenario3DiaSeguinte(encomendasExpressoNaoEntregues); //entregar as encomendas que restaram
}
```

Funcionalidade extra

Encomendas não entregues passam para o dia seguinte com prioridade

```
std::sort(encomendasDoDia.begin(), encomendasDoDia.end(),
          [](Encomenda e1, Encomenda e2){return e1.getDuracao() < e2.getDuracao();});//ordenar pela duracao

for (Encomenda &encomenda : encomendasDoDia) {
    encomendasDoDiaAnterior.push_back(encomenda); //juntar os dois vetores dando prioridade ao dia anterior
}
```

É utilizada uma função recursiva que enquanto existir encomendas que vieram de dias anteriores ela vai as processando de forma a dar prioridade às do dia anterior em relação às geradas para aquele dia e voltando a entregar todas as encomendas e verificar no final do dia se ainda sobraram algumas.

Solução algorítmica destacada

Decidimos destacar a solução gananciosa que utilizamos tanto no cenário 1 como no cenário 2 onde ordenamos tanto as carrinhas e as encomendas de acordo com as que queríamos que fossem utilizadas primeiro.

Assim conseguimos escolher a cada passo uma solução local ótima que nos daria uma solução global que apesar de poder não ser a ideal será sempre muito próxima de uma solução global ótima.

No geral, esta abordagem ,apesar de apenas dar uma solução muito próxima ideal, tem uma baixa complexidade temporal pelo o que permite dar uma solução mais rápida quando comparada a algoritmos com melhores soluções como brute force ou knapsack.

Principais Dificuldades

Principais dificuldades:

- Documentação do Doxygen muito trabalhosa
- Descobrir qual algoritmo aplicar em cada cenário

O esforço foi equivalente por todos os elementos do grupo.

Exemplos de execução

```
1 - Simular a entrega de encomendas normais
2 - Simular a entrega de encomendas expresso

1

1 - Minimizar o numero de estafetas usados
2 - Maximizar o lucro da empresa

1

Com quantas encomendas pretende simular?

80

Encomendas normais para este dia:

#################################################################################
#   ID     #     VOLUME     #     PESO     #    DURACAO    #   RECOMPENSA  #
#################################################################################
|   380    |        13      |       16     |      297      |     1785      |
+--------------------------------------------------------------------------------+
|   374    |        7       |       10     |      963      |     1630      |
+--------------------------------------------------------------------------------+
|   319    |        29      |       10     |      775      |      243      |
+--------------------------------------------------------------------------------+
|   105    |        9       |       19     |      241      |     1281      |
+--------------------------------------------------------------------------------+
|   442    |        10      |       10     |      631      |      553      |
+--------------------------------------------------------------------------------+
|   188    |        20      |       7      |      964      |      386      |
+--------------------------------------------------------------------------------+
```

```
Carrinhas usadas:

##############################
#   ID CARRINHA  :    1     #
#      LUCRO  :    7072     #
##############################
|   >3       >75      >265     |
|   >353     >245     >203     |
|   >44      >35      >100     |
|   >61      >433     >368     |
|   >425     >123     >310     |
+----------------------------+

##############################
#   ID CARRINHA  :   25     #
#      LUCRO  :    -249     #
##############################
|   >130     >298     >419     |
|   >131     >119     >341     |
|   >309     >243     >195     |
|   >11      >206     >43      |
|   >19      >190     >320     |
+----------------------------+
```

```
##############################
#   ID CARRINHA  :   16     #
#      LUCRO  :   10830     #
##############################
|   >319     >133     >175     |
|   >434     >386     >450     |
|   >82      >398     >183     |
|   >380     >2       >8       |
|   >176     >110     >105     |
|   >356     >429     >78      |
|   >188     >289     >247     |
|   >347     >406     >168     |
+----------------------------+

##############################
#   ID CARRINHA  :   12     #
#      LUCRO  :    8296     #
##############################
|   >46      >85      >442     |
|   >67      >229     >263     |
|   >374     >138     >197     |
|   >311     >361     >165     |
|   >439     >330     >420     |
|   >435     >22      >140     |
|   >344     >427     >392     |
+----------------------------+

Carrinha usadas: 4


Eficiencia: 100%
```

Exemplos de execução

```
1 - Simular a entrega de encomendas normais
2 - Simular a entrega de encomendas expresso

1

1 - Minimizar o numero de estafetas usados
2 - Maximizar o lucro da empresa

2

Com quantas encomendas pretende simular?

80

1 - Maximizar o lucro total (podendo ficar muitas encomendas por entregar)
2 - Maximizar o lucro total (entregando o maior numero de encomendas)

1

Encomendas normais para este dia:

################################################################################
#   ID     #     VOLUME     #     PESO     #     DURACAO     #   RECOMPENSA  #
################################################################################
|   125    |        10      |       26     |      237       |      1229      |
+--------------------------------------------------------------------------------+
|   193    |        13      |        3     |      818       |       203      |
+--------------------------------------------------------------------------------+
|   107    |        10      |        3     |      766       |       547      |
+--------------------------------------------------------------------------------+
```

```
Carrinhas usadas:

############################
#   ID CARRINHA  :   16    #
#      LUCRO     :  27853   #
############################
|   >149    >62     >25    |
|   >218    >207    >146   |
|   >315    >18     >210   |
|   >304    >107    >98    |
|   >187    >241    >132   |
|   >173    >297    >276   |
|   >148    >150    >347   |
|   >404    >280    >448   |
|   >273    >284    >339   |
|   >176    >57     >70    |
|   >274    >423    >193   |
+--------------------------+

############################
#   ID CARRINHA  :    1    #
#      LUCRO     :   9401   #
############################
|   >306    >398    >143   |
|   >151    >323    >361   |
|   >125    >355    >161   |
|   >136    >72     >10    |
|   >385    >419    >109   |
|   >170    >191    >127   |
+--------------------------+
```

```
############################
#   ID CARRINHA  :   25    #
#      LUCRO     :   4471   #
############################
|   >430    >169    >445   |
|   >19     >298    >386   |
|   >189    >369    >376   |
|   >158    >120    >192   |
|   >349    >309    >421   |
+--------------------------+

##############################
#  ID ENCOMENDAS NAO ENTREGUES  #
##############################
|            114             |
+----------------------------+
|            400             |
+----------------------------+
|             29             |
+----------------------------+
|            196             |
+----------------------------+
|            450             |
+----------------------------+
|            275             |
+----------------------------+
```

```
Lucro: 41725 euros

Eficiencia: 85%
```

Exemplos de execução

```
1 - Simular a entrega de encomendas normais
2 - Simular a entrega de encomendas expresso

1

1 - Minimizar o numero de estafetas usados
2 - Maximizar o lucro da empresa

2

Com quantas encomendas pretende simular?

80

1 - Maximizar o lucro total (podendo ficar muitas encomendas por entregar)
2 - Maximizar o lucro total (entregando o maior numero de encomendas)

2

Encomendas normais para este dia:

################################################################################
#   ID     #     VOLUME     #     PESO     #    DURACAO    #   RECOMPENSA  #
################################################################################
|   228    |        28      |       19     |      888      |      828      |
+--------------------------------------------------------------------------------+
|   437    |         7      |       10     |      919      |      801      |
+--------------------------------------------------------------------------------+
|    12    |        25      |       13     |      384      |     1251      |
+--------------------------------------------------------------------------------+
```

```
Carrinhas usadas:

##############################
#   ID CARRINHA  :   16      #
#      LUCRO     :   11957   #
##############################
|   >55      >35      >192    |
|   >318     >224     >86     |
|   >59      >36      >419    |
|   >350     >97      >317    |
|   >5       >353     >446    |
+---------------------------+

##############################
#   ID CARRINHA  :    1      #
#      LUCRO     :   10795   #
##############################
|   >10      >184     >323    |
|   >430     >230     >56     |
|   >216     >182     >349    |
|   >228     >227     >143    |
|   >12      >291     >159    |
|   >49      >183     >114    |
+---------------------------+
```

```
##############################
#   ID CARRINHA  :   25      #
#      LUCRO     :   2034    #
##############################
|   >288     >135     >131    |
|   >151     >258     >403    |
|   >364     >60      >357    |
|   >365     >205     >110    |
|   >129     >375     >448    |
|   >302     >113     >259    |
|   >210     >15      >18     |
+---------------------------+

##############################
#   ID CARRINHA  :   37      #
#      LUCRO     :   9314    #
##############################
|   >325     >315     >8      |
|   >146     >25      >254    |
|   >194     >22      >437    |
|   >174     >296     >439    |
|   >219     >226     >54     |
|   >9       >352     >91     |
|   >260     >395     >231    |
|   >141     >134     >262    |
+---------------------------+

Lucro: 34100 euros

Eficiencia: 100%
```

Exemplos de execução

```
1 - Simular a entrega de encomendas normais
2 - Simular a entrega de encomendas expresso

2

Com quantas encomendas pretende simular?

60


Encomendas expresso para este dia:

##############################################################################
#   ID     #     VOLUME     #     PESO     #    DURACAO    #   RECOMPENSA   #
##############################################################################
|   356    |        13      |       13     |      355      |      967      |
+------------------------------------------------------------------------------+
|   338    |        20      |       13     |      815      |     1054      |
+------------------------------------------------------------------------------+
|    15    |         9      |        9     |     1019      |     1137      |
+------------------------------------------------------------------------------+
|   103    |        16      |        6     |      874      |      755      |
+------------------------------------------------------------------------------+
```

```
Encomendas expresso entregues no dia de hoje:

###################################
#   HORA DE ENTREGA     #    ID    #
###################################
|        09:00          |   268    |
+-------------------------------------+
|        09:02          |   420    |
+-------------------------------------+
|        09:05          |   403    |
+-------------------------------------+
|        09:09          |   215    |
+-------------------------------------+
|        09:13          |   178    |
+-------------------------------------+
|        09:17          |   373    |
+-------------------------------------+
|        16:26          |   170    |
+-------------------------------------+
|        16:41          |   191    |
+-------------------------------------+
|        16:57          |    64    |
+-------------------------------------+

Numero de encomendas nao entregues : 4

Eficiencia: 93.3%

Press enter to continue...

Com quantas encomendas pretende simular?

30
```

```
----------DIA SEGUINTE----------

Encomendas expresso para este dia com prioridade:

##############################################################################
#   ID     #     VOLUME     #     PESO     #    DURACAO    #   RECOMPENSA   #
##############################################################################
|   337    |        26      |       26     |     1009      |     1261      |
+------------------------------------------------------------------------------+
|    15    |         9      |        9     |     1019      |     1137      |
+------------------------------------------------------------------------------+
|   408    |        22      |       28     |     1058      |      347      |
+------------------------------------------------------------------------------+
|   243    |        16      |       28     |     1094      |      875      |
+------------------------------------------------------------------------------+

Encomendas expresso para este dia:

##############################################################################
#   ID     #     VOLUME     #     PESO     #    DURACAO    #   RECOMPENSA   #
##############################################################################
|   174    |         3      |       19     |      540      |      952      |
+------------------------------------------------------------------------------+
|    88    |        13      |        3     |      937      |      466      |
+------------------------------------------------------------------------------+
|   313    |         3      |       19     |      980      |      901      |
+------------------------------------------------------------------------------+
```

```
Encomendas expresso entregues no dia de hoje:

###################################
#   HORA DE ENTREGA     #    ID    #
###################################
|        09:00          |   337    |
+-------------------------------------+
|        09:16          |    15    |
+-------------------------------------+
|        09:33          |   408    |
+-------------------------------------+
|        09:51          |   243    |
+-------------------------------------+
|        10:09          |   185    |
+-------------------------------------+
|        10:11          |   160    |
+-------------------------------------+
|        15:08          |   410    |
+-------------------------------------+
|        15:26          |   421    |
+-------------------------------------+

Eficiencia: 100%
```