

Trabalho Final Fundamentos de Projeto e Análise de Algoritmos

Integrantes:

Thiago Mariano
Felipe Espindola
Julia Evelyn
Octávio Rocha

O trabalho busca estudar e utilizar tipos de algoritmos estudados durante o semestre durante as aulas de Fundamento de Projeto e Análise de Algoritmo.

Problema 01 - Rotas de Caminhão:

Parâmetros utilizados como teste da modelagem e algoritmo desenvolvido:

Rotas (km) : 30 Km, 40 Km, 10 Km, 20 Km, 50 Km
Número de caminhões: 03 caminhões;

Algoritmo guloso

O algoritmo guloso é baseado na escolha rápida e máxima de preenchimento da resposta com o objeto atual, tendo seu limite de acordo com a lógica do problema. Dessa forma, como o problema não possui um teto máximo da rota de cada caminhão, o algoritmo irá preencher todo o primeiro caminhão com a quilometragem de todas as rotas, já que como dito anteriormente não há um gargalo que faça-o desistir dessa opção, porém como o objetivo é melhor distribuir as rotas entre os caminhões, **o algoritmo guloso sempre irá falhar na resolução desse problema.**

Tendo seu tempo médio de execução em 5 milésimos de segundo, já que não há consideração de looping e atribuições complexas dentro do algoritmo, tendo como complexidade (N) já que passar por todos os caminhões.

Segue execução final do código:

```
Caminhao: Caminhao1 total Km: 150  
Caminhao: Caminhao2 total Km: 0  
Caminhao: Caminhao3 total Km: 0  
Tempo de execucao: 5
```

Algoritmo BackTracking

O algoritmo BackTracking gera uma sequência de possibilidades de inserção dentro da solução, como uma árvore e em forma de busca em profundidade, podendo os ramos possíveis que ele visualizar que não irão agregar dentro da solução, o que evita diversos loopings que não vão agregar na solução e assim reduzimos a complexidade do algoritmo. No caso ele irá se basear unicamente em diminuir a diferença entre o caminhão com maior quilometragem de rota e o menor, logo ele irá verificar se colocar aquela rota no caminhão indicado no loop irá ajudar a diminuir a diferença, caso melhore a solução ele insere, caso não ele passa para o próximo caminhão, caso ele não encontre a possibilidade de melhorar ele irá inserir no caminhão de menor rota já existente

Em comparação com o método guloso se mostra bem mais eficiente já que agora conseguimos nos basear no **gargalo de manter dentro da menor diferença possível entre o de maior e menor quilometragem**, o que permite o algoritmo a solucionar o problema.

Tendo seu tempo médio de execução em **12 milésimo de segundos, o que já é considerável tendo em visto a pequena quantidade de elementos utilizados na questão, porém o resultado se mostra consistente e correto:**

```
Tempo de execucao: 12
Caminhao1: 80
Caminhao2: 80
Caminhao3: 71
Caminhao4: 64
Caminhao5: 100
```

Algoritmo Programação Dinâmica

O algoritmo de programação dinâmica utiliza de uma solução para problemas de subestrutura ótima, onde a solução do problema irá se basear em pequenos problemas onde sua solução irá compor a solução completa do problema. Primeiramente, o algoritmo define uma média para que todos os caminhões tentem se manter no mesmo número de quilômetros rodados. Porém, como o problema visa minimizar a distância percorrida entre eles, não significa que em todos os casos a distância será igual. O algoritmo utiliza de uma tabela para guardar os valores e vai aumentando a cada passo, aproveitando para reutilizar números que já foram utilizados. Então, o algoritmo começa fazendo a criação de uma tabela onde as linhas são os caminhões e as rotas são as colunas e nossa função é encontrar a maior quantidade de km estando dentro da média definida.

Exemplo: Rotas - 10Km, 20Km, 30Km, 40Km, 50 Km
Caminhões - 1,2,3

Média: $150 / 3 \Rightarrow$ Cada caminhão deve ter uma quantidade de km de aproximadamente 50 km.

O tempo gasto para a execução do algoritmo foi de **7.616ms** e deve se levar em conta que o algoritmo de programação dinâmica é recomendado para cenários que possuímos menos instâncias.

```
O caminhão 1 fará um trajeto pelas rotas 1-3-, com um valor total de 50 KM  
O caminhão 2 fará um trajeto pelas rotas 2-5-, com um valor total de 50 KM  
O caminhão 3 fará um trajeto pelas rotas 4-, com um valor total de 50 KM  
time: 7.616ms
```

Problema 02 - Acúmulo de Temperaturas:

Parâmetros utilizados como teste da modelagem e algoritmo desenvolvido:

Arquivos enviados: temperaturas.txt e temperaturasExtremo.txt

Algoritmo de Divisão e Conquista:

O algoritmo de divisão e conquista funciona baseando-se na quebra do problema em partes menores antes de começar a solucioná-lo, reduzindo significativamente a sua complexidade, o que o torna bastante eficiente.

No caso do problema de acúmulo de temperaturas, ele recebe os dados de temperaturas diárias de cada ano, e procura o intervalo de dias em que houve o maior acúmulo de temperatura. E por acúmulo de temperatura, pode se entender o período em que houve maior (1) soma das diferenças de temperatura entre cada um dos dias, ou a (2) soma das temperaturas propriamente ditas, que seria o mesmo que o período mais quente do ano, que houve o desgaste mais intenso nas geleiras. Para a 1ª interpretação, primeiro os dados são convertidos em um arranjo com a diferença entre a temperatura do dia $(i + 1)$ com a temperatura do dia (i) . O algoritmo desenvolvido lida com ambas as possibilidades de interpretação, sendo alteradas as comparações, na 1ª, de diferenças, e na 2ª, de temperaturas.

O primeiro passo é dividir o índice do arranjo por 2, até encontrar o menor índice possível, chegando ao caso base, quando o menor índice é igual ao maior, e armazená-lo no arranjo da esquerda, e em seguida encontrar o índice seguinte da direita, e armazená-lo no arranjo da direita, e após encontrar os índices menor, meio e maior da iteração atual, busca-se encontrar o maior subarranjo cruzado, entre os arranjos atuais da esquerda e direita. Finalmente é feita a comparação entre os valores encontrados das somas da esquerda, direita e cruzado.

Ele possui complexidade semelhante ao mergesort, sendo sua relação de recorrência $T(n) = 2T(n/2) + O(n)$, descrita no código, que solucionada, nos leva a um custo $n * \log(n)$. Tendo seu tempo médio de execução de 0,26 milissegundo que é bem rápido, considerando que a quantidade de dados não é tão pequena assim.

Após a resolução do problema principal, também foi desenvolvido um algoritmo para responder a necessidade dos pesquisadores de comparar as coincidências entre os períodos de ocorrência de desgaste, no qual é realizado a comparação dos períodos de maior desgaste de cada ano, e caso os dias de início ou fim sejam pertencentes ao intervalo entre o início e fim do ano comparado, uma coincidência é registrada, e é mantido o intervalo correspondente a interseção entre os dois intervalos.

Segue abaixo a execução final do código:

1. Para **soma das diferenças** de temperatura:

```
**** CONSIDERANDO ACÚMULO (SOMA) DE DIFERENCAS DE TEMPERATURA ENTRE OS DIAS ****

Período de Maior Acúmulo de Temperatura entre todos os anos {anoInicio: 1993, diaInicio: 195 ||
anoFim: 1994, diaFim: 6, maiorAcumulo: 44°C }

Ano 1993 - Período de Maior Acúmulo de Temperatura { inicio: 195, fim: 318, maiorAcumulo: 37°C }
Ano 1994 - Período de Maior Acúmulo de Temperatura { inicio: 185, fim: 305, maiorAcumulo: 37°C }
Ano 1995 - Período de Maior Acúmulo de Temperatura { inicio: 232, fim: 329, maiorAcumulo: 36°C }
Ano 1996 - Período de Maior Acúmulo de Temperatura { inicio: 194, fim: 314, maiorAcumulo: 37°C }
Ano 1997 - Período de Maior Acúmulo de Temperatura { inicio: 185, fim: 327, maiorAcumulo: 36°C }

COINCIDENCIAS:

Coincidencia { ano1: 1993, ano2: 1994, inicioCoincidencia: 195, fimCoincidencia: 305 }
Coincidencia { ano1: 1993, ano2: 1995, inicioCoincidencia: 232, fimCoincidencia: 318 }
Coincidencia { ano1: 1993, ano2: 1996, inicioCoincidencia: 195, fimCoincidencia: 314 }
Coincidencia { ano1: 1993, ano2: 1997, inicioCoincidencia: 195, fimCoincidencia: 318 }
Coincidencia { ano1: 1994, ano2: 1995, inicioCoincidencia: 232, fimCoincidencia: 305 }
Coincidencia { ano1: 1994, ano2: 1996, inicioCoincidencia: 194, fimCoincidencia: 305 }
Coincidencia { ano1: 1994, ano2: 1997, inicioCoincidencia: 185, fimCoincidencia: 305 }
Coincidencia { ano1: 1995, ano2: 1996, inicioCoincidencia: 232, fimCoincidencia: 314 }
Coincidencia { ano1: 1995, ano2: 1997, inicioCoincidencia: 232, fimCoincidencia: 327 }
Coincidencia { ano1: 1996, ano2: 1997, inicioCoincidencia: 194, fimCoincidencia: 314 }
```

2. Para soma das temperaturas propriamente ditas:

```
**** CONSIDERANDO ACÚMULO (SOMA) DE TEMPERATURAS PROPRIAMENTE DITAS ****
Período de Maior Acúmulo de Temperatura entre todos os anos {anoInicio: 1997, diaInicio: 9 ||
anoFim: 1997, diaFim: 18, maiorAcumulo: 42°C }

Ano 1993 - Período de Maior Acúmulo de Temperatura { inicio: 24, fim: 30, maiorAcumulo: 26°C }
Ano 1994 - Período de Maior Acúmulo de Temperatura { inicio: 23, fim: 31, maiorAcumulo: 27°C }
Ano 1995 - Período de Maior Acúmulo de Temperatura { inicio: 24, fim: 32, maiorAcumulo: 33°C }
Ano 1996 - Período de Maior Acúmulo de Temperatura { inicio: 28, fim: 31, maiorAcumulo: 36°C }
Ano 1997 - Período de Maior Acúmulo de Temperatura { inicio: 9, fim: 18, maiorAcumulo: 42°C }

COINCIDENCIAS:

Coincidencia { ano1: 1993, ano2: 1994, inicioCoincidencia: 24, fimCoincidencia: 30 }
Coincidencia { ano1: 1993, ano2: 1995, inicioCoincidencia: 24, fimCoincidencia: 30 }
Coincidencia { ano1: 1993, ano2: 1996, inicioCoincidencia: 28, fimCoincidencia: 30 }
Coincidencia { ano1: 1994, ano2: 1995, inicioCoincidencia: 24, fimCoincidencia: 31 }
Coincidencia { ano1: 1994, ano2: 1996, inicioCoincidencia: 28, fimCoincidencia: 31 }
Coincidencia { ano1: 1995, ano2: 1996, inicioCoincidencia: 28, fimCoincidencia: 31 }
```

É possível concluir que a solução se mostra eficaz e fornece as respostas pedidas pelos pesquisadores.