Problema B

Thumbnails

Problema

Considere o problema da representação e manipulação eficiente de imagens quadradas.

Uma imagem quadrada, na sua forma básica, é uma matriz n por n de pixeis, em que n é o tamanho da imagem e um píxel é a unidade básica (um ponto) com a informação da cor. Neste exercício vamos considerar que n é uma potencia de dois (1, 2, 4, 8, 16 etc.) e que cada pixel tem uma de duas cores : preta ou branca.

Para a manipulação de uma imagem, é muitas vezes mais cómodo e eficiente usar representações alternativas à representação matricial. É o que faremos neste exercício. Usaremos árvores quaternárias *quadtrees* para representar estas imagens. Este tipo de árvore está codificado no tipo OCaml `image` apresentado a seguir.

```
type color = W | B (* W: White, B: Black *)
type image = L of color (* leaf of one color *)
           | N of image * image * image * image  (* node with four
             children *)
```

Este formato tem a vantagem de poder compactar a representação de uma imagem (num tamanho menor do que a matriz subjacente) tirando proveito de padrões cromáticos presentes na imagem. A definição do tipo aqui dada o é a título de exemplo. Este poderá ser alterado conforme eventuais necessidades.

A construção da árvore a partir da matriz é feita de forma recursiva. Uma imagem (matriz) é dividida *ordenadamente* em 4 partes iguais. As partes `NW`, `NE`, `SE` e `SW`, como o mostra a figura 1

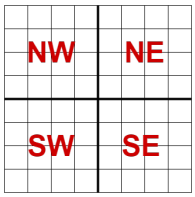

Figura 1: divisão em 4 partes

Aplica-se recursivamente esta subdivisão à cada parte que não seja de uma só cor.

Assim, uma imagem como a figura 2 subdivide-se nas seguinte sub-imagens (cada sub-imagem

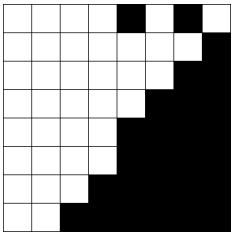

Figura 2: uma imagem subdividida - 1

não cromaticamente constante é sujeita a uma subdivisão) conforme as figuras 3 4 e 5. e a árvore resultante é apresentada na figura 6 onde, tendo em conta o tipo `imagem`, os nós cinzentos são os nós `N`, as folhas brancas são os elementos `L W` e as folhas pretas são os elementos `L B`. Guardamos as indicações `NW`, `NE`, `SE`, `SW` para fins de clarificação da construção da árvore.

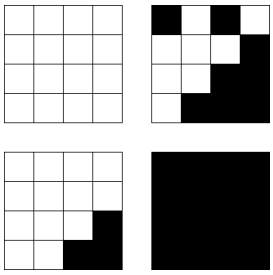

Figura 3: uma imagem subdividida - 2

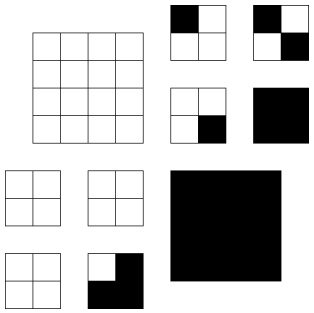

Figura 4: uma imagem subdividida - 3

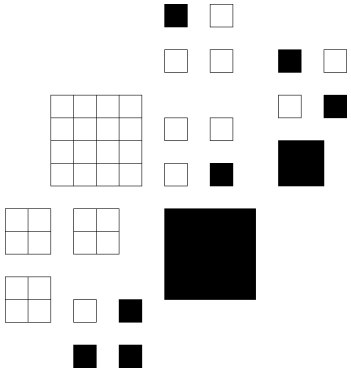

Figura 5: uma imagem subdividida - 4 e fim

Consideramos agora o problema da criação de *thumbnail*. Um *thumbnail* é uma imagem de modesta dimensão que resuma outra imagem (maior).

A construção de um *thumbnail* numa imagem representada por uma `quadtree` é muito simples.

Se pretendemos um *thumbnail* de tamanho p de uma imagem $n \times n$, com $p \leq n$ e p potência de dois, **basta cortar a `quadtree` na altura certa**.

Este corte deixará as folhas originais que existam até este nível e criará novas folhas quando efectivamente se tiver de descartar sub-árvores da árvore original. O corte resulta numa folha cuja cor se estabelece da seguinte forma:

- calcular a quantidade de pixeis brancos abrangidos pela sub-árvore em questão, digamos q_w;
- calcular a quantidade de pixeis pretos abrangidos pela sub-árvore em questão, digamos q_b;
- se $p_b \geq p_w$, a folha resultante é preta, senão é branca.

Por exemplo, na figura 7, pretendemos um *thumbnai*l de 4 por 4 ($p = 4$) a partir da imagem 8 por 8 ($n = 8$) previamente apresentada. O corte terá lugar na linha tracejada vermelha. O efeito

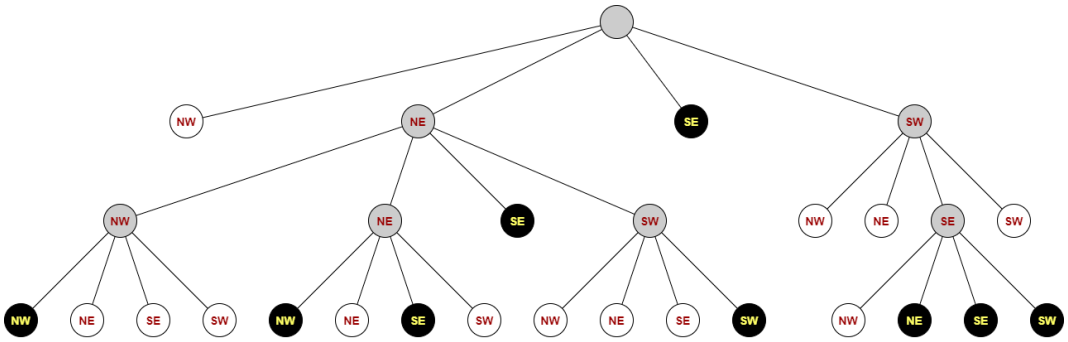

Figura 6: a árvore para a imagem da figura 2

será então cortar 4 sub-árvores e substitui-las por 4 folhas novas. Cada uma das folhas novas é um resumo da árvore que está a substituir. Como tal está destacada com um círculo vermelho e substituirá a sub-árvore cuja raiz está imediatamente ao lado. A cor desta folha é calculada de acordo com os pixeis contidos na sub-árvore assinalada pelo triângulo vermelho subjacente.

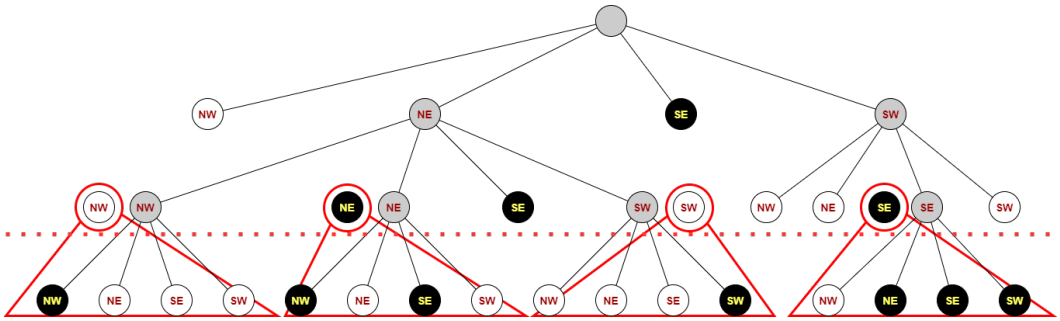

Figura 7: construção do thumbnail

Listemos agora os diferentes cálculos requeridos para o exercício.

Primeiro, é esperado que saibam construir uma *quadtree* a partir de uma matriz dada na forma de um ficheiro `pbm` ASCII e sem comentários (no formato `P1`, ver (*pbm* na Wikipédia).

Segundo, é esperado que saibam determinar qual é a profundidade da (das?) folha mais alta na árvore resultante. É 1 no exemplo dado.

Terceiro, é esperado que saibam determinar quantas folhas a *quadtree* contém. É 22 no exemplo dado.

Finalmente, dado o tamanho do *thumbnail*, é esperado que calculem a *quadtree* do *thumbnail* requerido e que o devolvam na saída standard na sua forma matricial.

Entrada

A entrada começa pela especificação de uma imagem no formato ppm sem comentários A linha final contém o inteiro p, potência de 2, que indica o tamanho do *thumbnail* por calcular.

Saída

Uma linha com o valor inteiro que indica a profundidade da folha mais alta da árvore calculada. Uma linha que indica o número de folhas totais da árvore. Uma matriz p por p que contém o *thumbnail* calculado. Esta matriz está organizada em p linhas de p inteiros

Limites

Os valores de n e p são potências de dois. É garantido que $0 < p \leq n \leq 1024$.

Exemplo de Entrada

```
P1
8 8
0 0 0 0 1 0 1 0
0 0 0 0 0 0 0 1
0 0 0 0 0 0 1 1
0 0 0 0 0 1 1 1
0 0 0 0 1 1 1 1
0 0 0 0 1 1 1 1
0 0 0 1 1 1 1 1
0 0 1 1 1 1 1 1
4
```

Exemplo de Saída

```
1
22
0 0 0 1
0 0 0 1
0 0 1 1
0 1 1 1
```