

Problema C: com imagens é melhor

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 pixels, 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 potência 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, *quadrees*, 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 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

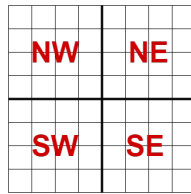


Figura 1: divisão em 4 partes

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

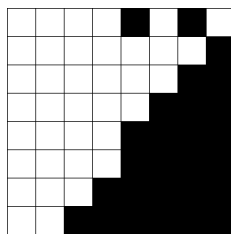


Figura 2: uma imagem subdividida - 1

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 `image`, 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.

Listemos agora os diferentes cálculos requeridos para o exercício.

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

Segundo, é esperado que saibam determinar o número de folhas L e o número de nós N da árvore *quadree* resultante da leitura de uma imagem em entrada. No exemplos dado, são 22 folhas e 7 nodos internos.

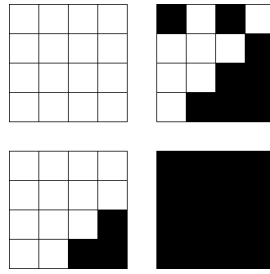


Figura 3: uma imagem subdividida - 2

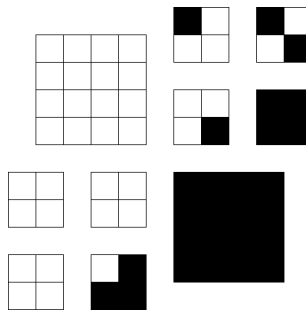


Figura 4: uma imagem subdividida - 3

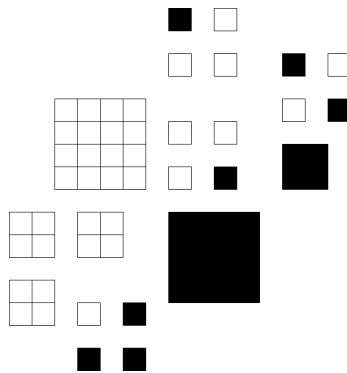


Figura 5: uma imagem subdividida - 4 e fim

Terceiro, é esperado que saibam determinar o tamanho do ramo mais curto e o do ramo mais longo. Na *quadtree* do exemplo, estes valores são 1 e 3.

Finalmente, pretendemos que saiba fazer

- uma rotação da imagem de 90 graus para a esquerda e mostrar a imagem resultante;
- em seguida inverter a imagem (os pixels brancos passam a preto e os pixels pretos) e mostrar ao resultado;
- em seguida uma rotação de 180 graus para a direita e mostrar ao resultado.

Dica: O tipo *image* acima descrito pode ser estendido conforme as necessidades. Por exemplo, pode ser aumentado para conter mais informação ajudando assim a um mais fácil processamento da informação tratada.

As soluções que não operam sob *quadtrees* serão rejeitadas pela equipa docente

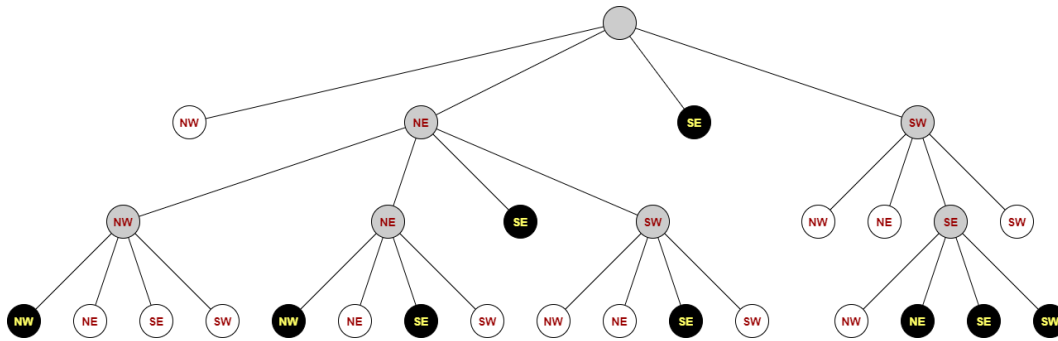


Figura 6: a árvore para a imagem da figura 2

Entrada

A entrada começa pela especificação de uma imagem no formato ppm sem comentários. Neste formato, o tamanho da matriz quadrado, n , é dado na segunda linha em duplicado (porque é uma imagem quadrada).

Saída

Uma linha com dois valores inteiros (separados por um espaço) que indica o número de folhas e de nós internos da *quadtree* construída.

Uma segunda linha com dois inteiros (separados por um espaço) dando conta do tamanho mínimo e máximo dos ramos da árvore.

Três imagens no formato ppm sem comentários, que resultam das três operações requeridas.

Limites

Os valores de n e p são potência 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 0 1 1 1 1
0 0 0 1 1 1 1 1
0 0 1 1 1 1 1 1
```

Exemplo de Saída

```
22 7
1 3
P1
8 8
0 1 1 1 1 1 1 1
1 0 1 1 1 1 1 1
0 0 0 1 1 1 1 1
1 0 0 0 1 1 1 1
```

0 0 0 0 0 0 1 1
0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0

P1

8 8

1 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0
1 1 1 0 0 0 0 0
0 1 1 1 0 0 0 0
1 1 1 1 1 1 0 0
1 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1

P1

8 8

1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
0 1 1 1 1 1 1 1
0 0 1 1 1 1 1 1
0 0 0 0 1 1 1 0
0 0 0 0 0 1 1 1
0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 1