

Elementos de Programação - IST - LMAC, MEBiom

21 de Dezembro de 2018

Ficha 3A

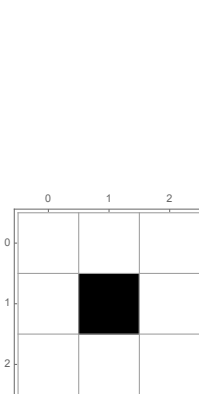
Duração: 15m

Número: _____ Nome: _____

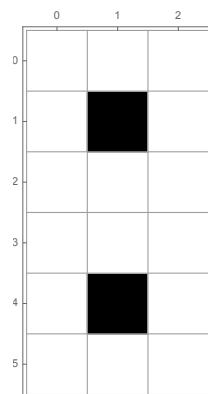
Considere *bitmaps* (figuras rectangulares, com *pixels* a preto-e-branco), com as operações:

- `white(n,m)`: bitmap com n linhas, m colunas e todos os pixels em branco;
 - `flip(b,i,j)`: bitmap que resulta de trocar a cor do pixel correspondente à linha i coluna j do bitmap b ;
 - `isblack(b,i,j)`: True se é preto o pixel correspondente à linha i coluna j do bitmap b , e False caso contrário;
 - `width(b)`: largura do bitmap b ;
 - `height(b)`: altura do bitmap b ;
 - `vglue(b1,b2)`: bitmap que resulta de colar verticalmente o bitmap $b1$ acima do bitmap $b2$, assumindo que os bitmaps têm a mesma largura.
- a) Em *Python*, pretende-se representar cada bitmap como uma matriz de entradas 0,1 (0 para pixels a branco e 1 para pixels a preto). Apresente implementações eficientes apenas para as operações `white`, `flip` e `vglue`.
- b) Desenvolva, sobre a camada de abstracção obtida acima e assegurando a independência da implementação, uma função `hglue` que recebendo bitmaps $b1,b2$, com a mesma altura, devolve o bitmap que resulta de colar horizontalmente $b1$ à esquerda de $b2$.

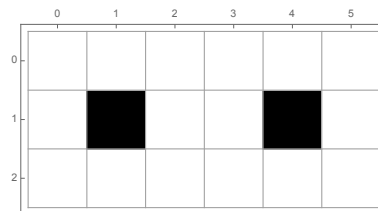
Abaixo representa-se graficamente a figura correspondente ao bitmap `fig` resultante de `flip(white(3,3),1,1)`, bem como os bitmaps que resultam de `vglue(fig,fig)` e `hglue(fig,fig)`.



`fig`



`vglue(fig,fig)`



`hglue(fig,fig)`

(vsff)

Elementos de Programação - IST - LMAC, MEBiom

21 de Dezembro de 2018

Ficha 3B

Duração: 15m

Número: _____ Nome: _____

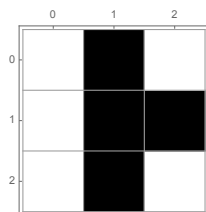
Considere *bitmaps* (figuras rectangulares, com *pixels* a preto-e-branco), com as operações:

- `white(n,m)`: bitmap com n linhas, m colunas e todos os pixels em branco;
- `flip(b,i,j)`: bitmap que resulta de trocar a cor do pixel correspondente à linha i coluna j do bitmap b ;
- `color(b,i,j)`: "black" se é preto o pixel na linha i coluna j do bitmap b , e "white" caso contrário;
- `negative(b)`: bitmap que resulta de trocar a cor a todos os pixels de b ;
- `dim(b)`: dimensões (largura e altura) do bitmap b ;
- `moveup(b)`: bitmap que resulta de deslocar cada pixel do bitmap b para a linha imediatamente acima, ficando a última linha em branco (os pixels da primeira linha de b desaparecem).

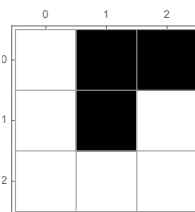
a) Em *Python*, pretende-se representar cada bitmap como um triplo da forma $(n,m,blacks)$ onde n é o número de linhas, m é o número de colunas, e $blacks$ é a lista das posições (i,j) dos pixels a preto. Apresente implementações eficientes apenas das operações `white`, `color` e `moveup`.

b) Desenvolva, sobre a camada de abstracção obtida acima e assegurando a independência da implementação, uma função `moveright` que recebendo um bitmap b devolve o bitmap que resulta de deslocar cada pixel do bitmap b para a coluna imediatamente à direita, ficando a primeira coluna em branco (os pixels da última coluna de b desaparecem).

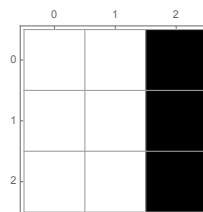
Abaixo representa-se graficamente a figura correspondente ao bitmap `fig` resultante de `flip(flip(flip(flip(white(3,3),0,1),1,1),1,2),2,1)`, bem como os bitmaps que resultam de `moveup(fig)` e `moveright(fig)`.



`fig`



`moveup(fig)`



`moveright(fig)`

(vsff)

Elementos de Programação - IST - LMAC, MEBiom

21 de Dezembro de 2018

Ficha 3C

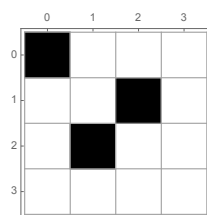
Duração: 15m

Número: _____ Nome: _____

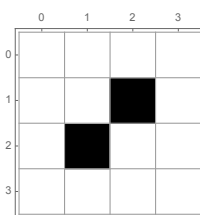
Considere *bitmaps* (figuras rectangulares, com *pixels* a preto-e-branco), com as operações:

- `black(n,m)`: bitmap com n linhas, m colunas e todos os pixels em preto;
 - `flip(b,i,j)`: bitmap que resulta de trocar a cor do pixel da linha i coluna j do bitmap b ;
 - `iswhite(b,i,j)`: `True` se é branco o pixel correspondente à linha i coluna j do bitmap b , e `False` caso contrário;
 - `negative(b)`: bitmap que resulta de trocar a cor a todos os pixels de b ;
 - `dim(b)`: dimensões (largura e altura) do bitmap b ;
 - `wframe(b)`: bitmap que resulta de tornar brancos todos os pixels exteriores (na primeira e última linha ou coluna) de b .
- a) Em *Python*, pretende-se representar cada bitmap como uma matriz de entradas booleanas (`True` para pixels a preto e `False` para pixels a branco). Apresente implementações eficientes apenas para as operações `black`, `negative` e `wframe`.
- b) Desenvolva, sobre a camada de abstracção obtida acima e assegurando a independência da implementação, uma função `bframe` que recebendo um bitmap b devolve o bitmap que resulta de tornar pretos todos os seus pixels exteriores (na primeira e última linha ou coluna).

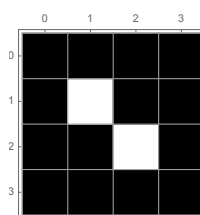
Abaixo representa-se graficamente a figura correspondente ao bitmap `fig` resultante de `flip(flip(flip(negative(black(4,4)),0,0),1,2),2,1)`, bem como os bitmaps que resultam de `wframe(fig)` e `bframe(fig)`.



`fig`



`wframe(fig)`



`bframe(fig)`

(vsff)

Elementos de Programação - IST - LMAC, MEBiom

21 de Dezembro de 2018

Ficha 3D

Duração: 15m

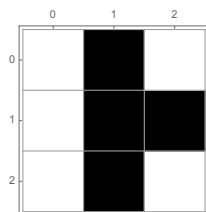
Número: _____ Nome: _____

Considere *bitmaps* (figuras rectangulares, com *pixels* a preto-e-branco), com as operações:

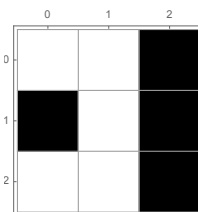
- `black(n,m)`: bitmap com n linhas, m colunas e todos os pixels em preto;
- `flip(b,i,j)`: bitmap que resulta de trocar a cor do pixel correspondente à linha i coluna j do bitmap b ;
- `color(b,i,j)`: "white" se é branco o pixel na linha i coluna j do bitmap b , e "black" caso contrário;
- `negative(b)`: bitmap que resulta de trocar a cor a todos os pixels de b ;
- `dim(b)`: dimensões (largura e altura) do bitmap b ;
- `scrollright(b)`: bitmap que resulta de deslocar cada pixel do bitmap b para a coluna imediatamente à direita, passando os pixels da última coluna de b para a primeira coluna.

- a) Em *Python*, pretende-se representar cada bitmap como um triplo da forma $(n,m,whites)$ onde n é o número de linhas, m é o número de colunas, e $whites$ é a lista das posições (i,j) dos pixels a branco. Apresente implementações eficientes apenas das operações `black`, `color` e `scrollright`.
- b) Desenvolva, sobre a camada de abstracção obtida acima e assegurando a independência da implementação, uma função `scrolldown` que recebendo um bitmap b devolve o bitmap que resulta de deslocar cada pixel do bitmap b para a linha imediatamente abaixo, passando os pixels da última linha de b para a primeira linha.

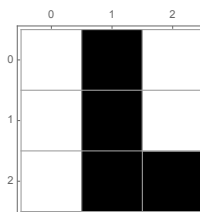
Abaixo representa-se a figura correspondente ao bitmap `fig` resultante de `flip(flip(flip(flip(negative(black(3,3))),0,1),1,1),1,2),2,1)`, e também bitmaps que resultam de `scrollright(fig)` e `scrolldown(fig)`.



`fig`



`scrollright(fig)`



`scrolldown(fig)`

(vsff)