Trabalho I

Pesca Pirata

Destemidos marujos navegam nos 7 mares buscando glória, alimento e riquezas. Cada nau pode se deslocar, arremessar redes e disparar canhões. Cada um destes movimentos provoca o gasto de uma certa quantidade de energia ou pode resultar na obtenção de energia extra. O gasto de energia é proporcional ao "esforço" realizado, dado pelas fórmula abaixo. Considere que *u* seja a unidade de medida; *d*, de deslocamento; *A*, da área da rede.

Deslocamento	$E_d = \frac{d}{5}$
Arremesso	$E_d = \frac{A}{25} * \frac{d}{5}$
Tiro	$E_t = d$

Uma rede é lançada ao norte, sul, leste ou oeste da embarcação e pode capturar moedas de ouro, lagostas, camarões e peixes, além de algas e outros materiais que são descartados. Cada moeda capturada incrementa o nível de energia da nau e os outros frutos do mar serão vendidos ao final da pesca e resultarão em riquezas[1] para os marujos, conforme a tabela abaixo:

Moeda	Cada moeda incrementa a energia em 0.5 unidades
Lagosta	Cada: M$ 20,00
Camarão	Cada: M$ 1,00
Peixe	Cada: M$ 5,00

Um tiro de canhão é disparado ortogonalmente a alguma lateral da nau. Caso atinja outra nau, a nau atingida é destruida e sua riqueza é capturada pela agressora.

A Entrada

A entrada do algoritmo será basicamente um conjunto de formas geométricas básicas (retângulos, círculos, etc) dispostos numa região do plano cartesiano que representam as entidades da pescaria e também um arquivo de consultas que "provocam os movimentos" da pescaria.

Arquivo .geo

Considere a Ilustração 1. Cada forma geométrica é definida por uma coordenada âncora

1 A unidade de riqueza (i.e., dinheiro) é a Merreca (M$).

(marcada, na figura, por um pequeno ponto vermelho) e por suas dimensões. A coordenada âncora do círculo é o seu centro e sua dimensão é definida por seu raio (r, na figura). A coordenada âncora do retângulo é seu canto inferior esquerdo[2] e suas dimensões são sua largura (w) e sua altura (h). A coordenada âncora de um texto, normalmente, é o início do texto, porém, pode ser definida como o meio ou o fim do texto. Por fim, uma linha é determinada por duas âncoras em suas extremidades. As coordenadas que posicionam as formas geométricas são valores reais.

Cada forma geométrica é identificada por um número inteiro.

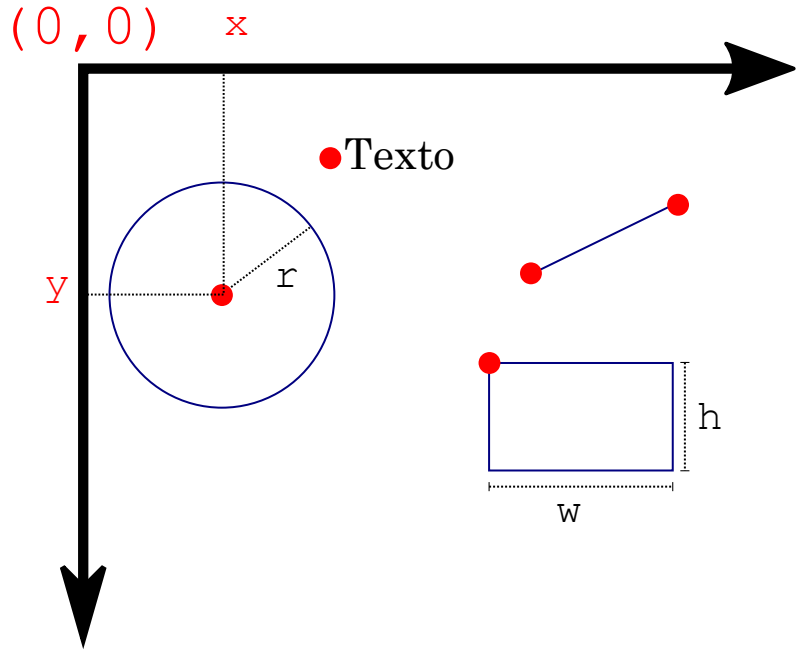


Ilustração 1: Formas no plano

As tabelas abaixo mostram os formatos dos arquivos de entrada (.geo e .qry). Cada comando tem um certo número de parâmetros. Os parâmetros mais comuns são:

- `i,j,k`: número inteiro, maior ou igual a 1. Identificador de uma forma geométrica.
- `r`: número real. Raio do círculo.
- `x,y`: números reais. Coordenada (x,y).
- `cor`: string. Cor válida dentro do padrão SVG.[3]

Alguns comandos utilizam memória auxiliar para armazenar identificadores

comando	parâmetros	descrição
c	`i x y r corb corp`	*desenhar círculo.* corb *é a cor da borda e* corp *é a cor do preenchimento*
r	`i x y w h corb corp`	*desenhar retângulo: w é a largura do retângulo e h, a altura.* corb *é a cor da borda e* corp *é a cor do preenchimento*

2 Note que o plano cartesiano está desenhado "de ponta-cabeça" em relação à representação usual.

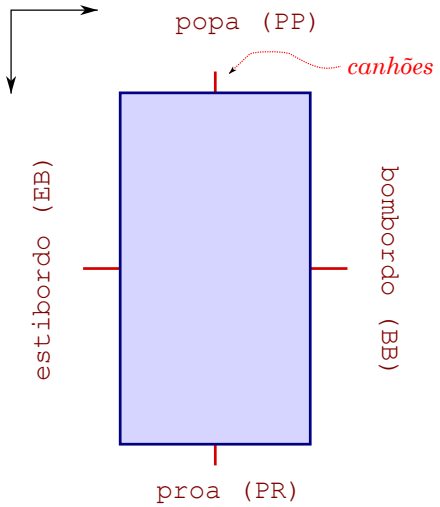
3 http://www.december.com/html/spec/colorsvg.html.
https://www.w3.org/Graphics/SVG/IG/resources/svgprimer.html

l	`i x1 y1 x2 y2 cor`	*Desenhar linha com extremidades nos pontos (x1,y1) e (x2,y2), com a cor especifiicada.*
t	`i x y corb corp a txto`	*desenha o texto* txto *nas coordenadas (**x,y**) e com a cores indicadas.* corb *é a cor da borda e* corp *é a cor do preenchimento. O parâmetro* **a** *determina a posição da âncora do texto:* **i**, *no início;* **m**, *no meio,* **f**, *no fim.* *O texto txto é o último parâmetro do comando. Pode incluir espaços em branco e se estende até o final da linha.*
comandos .geo		

Os elementos acima representam naus, lagostas, etc, como descrito abaixo:

retângulos	Naus
texto >-|-<	Lagosta
texto $	Moeda
qualquer outro texto	Algas, detritos, ...
círculo	Peixe
linha	Camarão

Considere a figura abaixo. Os lados de uma nau são identificados pelas siglas PP (popa, menor coordenada Y), PR (proa, maior Y), EB (estibordo, menor X), BB (bombordo, maior X). Uma nau possui canhões nos pontos médios de cada um de seus lados



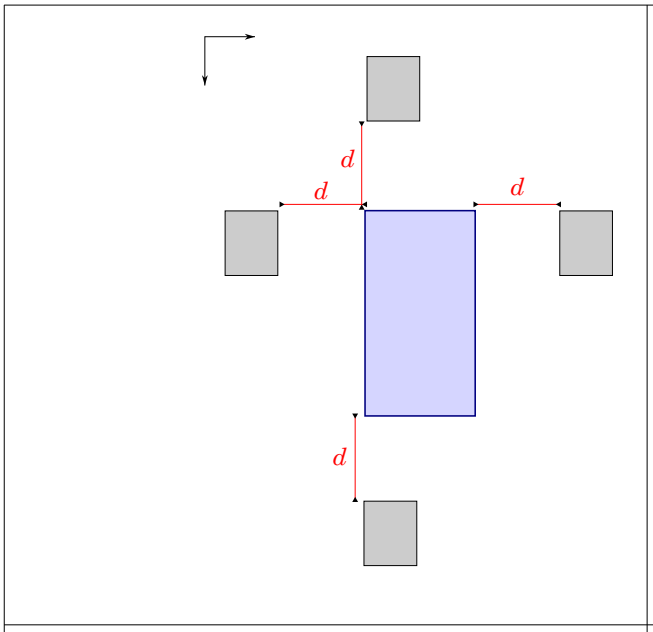
Arquivo .qry

No arquivo .qry estão as ações da pescaria. Os principais parâmetros destes comandos são:

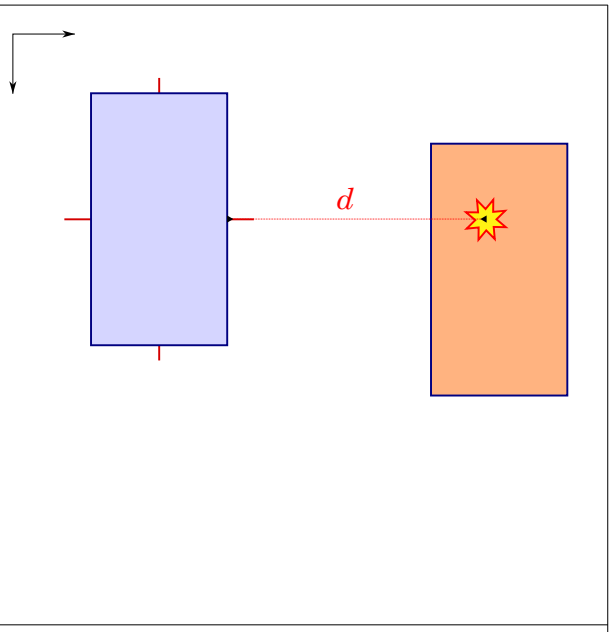
- x,y: (valores reais) ponto (x,y)
- w,h: (valores reais, não negativos) respectivamente, largura (projeção em X) e altura (projeção em Y) de um retângulo ou de uma região retangular
- lado: lado da nau: BB, EB, PP, PR.
- dx, dy: (valores reais, podem ser negativos).
- i: (valor inteiro) identificador da forma geométrica
- d: (valor real não negativo): distância

comando	parâmetros	descrição
e	v	*Energiza todas as naus com o nível v.* *TXT. Reportar os identificadores e dados dos retângulos (que representam as naus)*
mv	i dx dy	*Desloca a forma de identificador i de dx no eixo x e dy no eixo y.* *TXT: reportar os dados da forma de identificador i, incluindo a posição inicial e a posição final.* *SVG: Naturalmente, a forma é mostrada na posição final.*
lr	i lado d w h	*Lança rede de dimensão (w,h) a uma distância d do lado especificado da nau de identificador i, se a nau tiver energia suficiente. Os elementos capturados são contabilizados e removidos* *SVG: desenhar região onde a rede caiu.* *TXT: reportar os dados dos elementos capturados, incluindo seus valores monetários, o total desta captura, e a soma de todas as capturas, a energia antes e depois do lançamento.*
d	i lado d	*Dispara canhão da nau i do lado especificado, se a nau tiver energia suficiente. A carga atinge a distância d. Se a carga atingir uma nau, esta nau é destruida (como especificado na descrição)* *SVG: colocar um asterico no ponto de impacto. Naturalmente, a nau destruida não aparecerá na saida final.* *TXT: reportar o ponto de impacto. Caso uma nau for atingida, reportar também os dados da nau.*
mc	dx dy x y w h	*Translada os peixes que estiverem dentro da região (x,y,w,h) de dx e dy.* *TXT: reporta os dados das figuras transladadas, incluindo a posição inicial e final.*
Comandos .qry		

As figuras abaixo exemplificam lançamentos de redes e disparos de canhões.



*Lançamento da rede a uma distância **d** a partir de cada um dos lados da nau.*



*Disparo do canhão de bombordo (BB) com alcance **d**. Similar para os outros 3 canhões.*

A Saída

A saída deve estar de acordo com a descrição acima. Ao final do processamento do arquivo .qry, as figuras que não foram removidas naturalmente são mostradas no svg final e o txt apresenta a contabilidade final da pescaria, por nau.´Também, deve ser produzido um arquivo .dot que "desenha" o estado final da árvore Rubro-Negra. Os nós da árvore devem estar pintados de vermelho ou preto.

As naus também devem ter contornos grossos e pintados de acordo com seus níveis de energia final, conforme a tabela abaixo:

código cor		nível de energia
#484537		0
#FFCC00		0 < e < 10
#217821		10 <= e < 30
#800066		30 <= e

IMPLEMENTAÇÃO

As formas devem ser armazenadas em árvore Rubro-Negra, de acordo com especificado no arquivo .h fornecido.

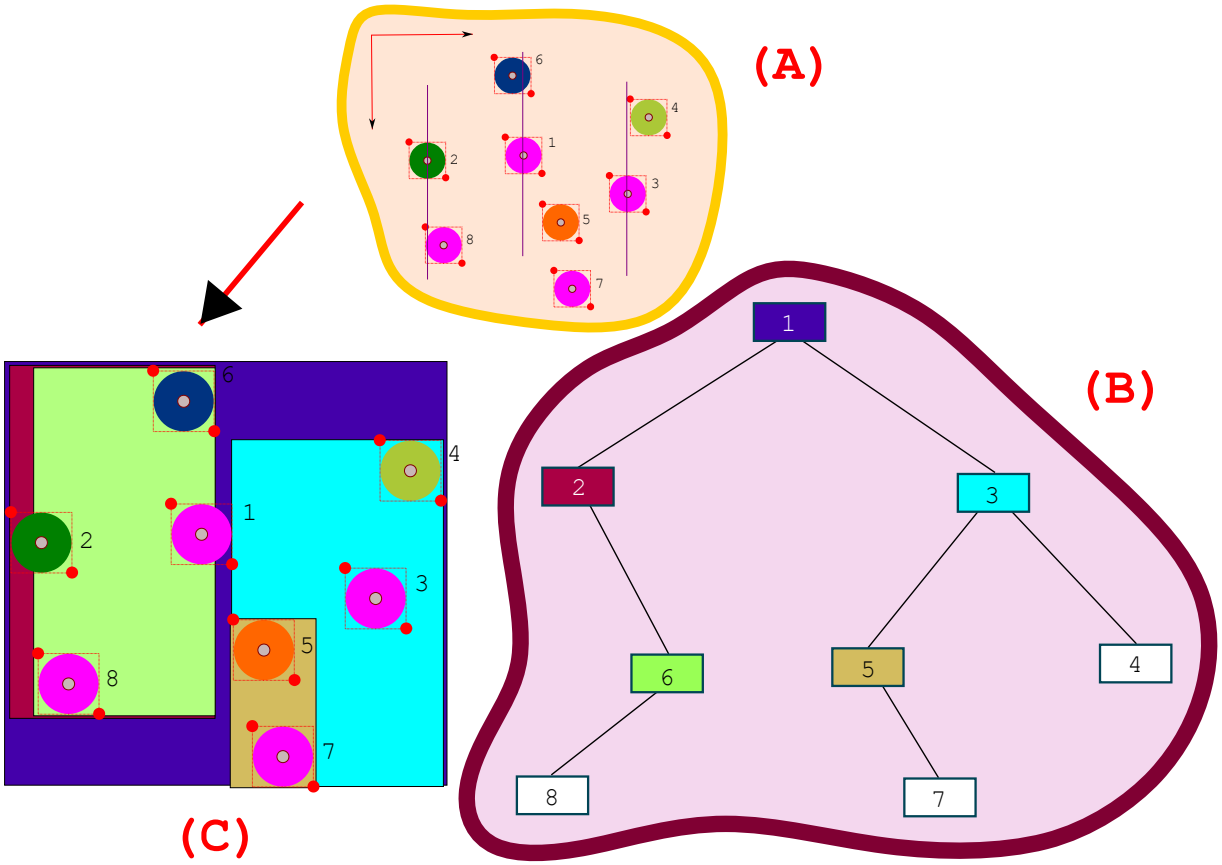
IMPORTANTE: as buscas por região na árvore **devem** "podar" ramos que, com certeza, não conterão nós dentro da região.

Os nós da árvore, além das informações especificadas no arquivo .h, **devem** armazenar o retângulo que envolve as informações do referido nó e dos nós de suas sub-árvores. A figura abaixo mostra um exemplo. Veja a região (A). As informações, neste exemplo, são círculos. Cada círculo

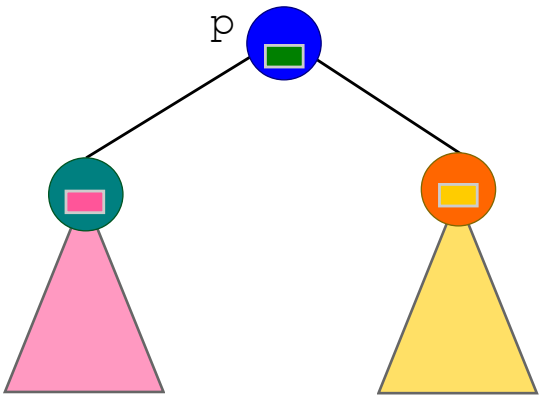
possui seu respectivo retângulo envolvente e sua âncora. Os círculos estão numerados. Na região (B) está uma árvore Binária de Busca resultante da inserção dos círculos na ordem dos identificadores. A região (C) é a região (a) levemente ampliada.

Note que o nó 6 é pai do nó 8. O menor retângulo envolvente que alberga o nó 6 e todos os seus descendentes está pintado em (C) com a mesma cor do nó 6 (verde claro).

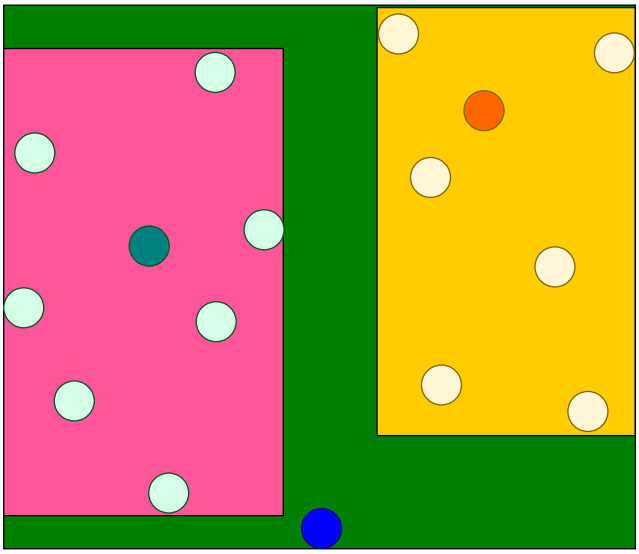
Note, também, os retângulos que envolvem o nó 5 e seus descendente (marrom) e o nó 3 e seus descendentes (azul claro). Note que o retângulo azul claro também engloba os nós 5 e 7.



Calcular o menor retângulo envolvente de uma árvore enraizada num nó p é bastante fácil (veja a figura abaixo). Basta considerar o retângulo envolvente da sub-arvore esquerda, da sub-árvore direita e o retângulo envolvente da informação armazenada no próprio nó p.



mbb(p) = mbb(p.esq) U mbb(p.dir)



É **terminantemente proibido** declarar structs nos arquivos de cabeçalho (.h).

O programa deve estar bem modularizado (arquivos .h e .c). Cada estrutura de dados deve estar em um módulo separado. O arquivo .h deve estar muito bem documentado (lembre-se que é um **"contrato"**).

AVALIAÇÃO

Espera-se uma atitude pró-ativa para a aquisição dos conhecimentos (i.e., estudo) para resolver o problema proposto.

A avaliação consistirá da execução dos testes e da inspeção de código. Além disso, deverá ser produzido um vídeo de, aproximadamente, 5 minutos no qual apresenta o programa. O aluno deve ter por objetivo convencer o avaliador que: (a) o programa funciona; (b)
o programa foi bem implementado O vídeo deve ser colocado no Youtube e o seu link deve estar anotado no arquivo leiame.

O Que Entregar

Submeter no Classroom o arquivo .zip com os fontes , conforme descrito anteriormente.

RESUMO DOS PARÂMETROS DO PROGRAMA TED

Parâmetro / argumento	Opcional	Descrição
`-e` ***`path`***	S	Diretório-base de entrada (**BED**)
`-f` ***`arq`*****`.geo`**	N	Arquivo com a descrição da cidade. Este arquivo deve estar sob o diretório **BED**.
`-o` ***`path`***	N	Diretório-base de saída (**BSD**)
`-q` ***`arqcons`*****`.qry`**	S	Arquivo com consultas. Este arquivo deve estar sob o diretório **BED**.

RESUMO DOS ARQUIVOS PRODUZIDOS

-f	-q	comando com sufixo	arquivos
arq.geo			arq.svg
arq.geo	***arqcons***.qry		arq.svg arq-arqcons.svg arq-arqcons.txt
arq.geo	***arqcons***.qry	***sufx***	arq.svg arq-arqcons.svg arq-arqcons.txt arq-arqcons-sufx.[svg|txt][4]

ATENÇÃO:
* os fontes devem ser compilados com a opção `-fstack-protector-all`.
* adotamos o padrão C99. Usar a opção `-std=c99`.

4 Podem ser produzidos os respectivos arquivos .svg e/ou .txt, dependendo da especificação do comando.