

# Trabalho II<sup>1</sup>

**FARMKRAFT** é um jogo de simulação agrícola. Em uma fazenda são plantados morangos, cebolas, repolhos, cenouras e abóboras, além das eventuais ervas daninhas. A lavoura pode ser atacada por pragas, pode ser tratada com defensivos e pode ser melhorada com adubos. Eventualmente, as plantações são colhidas e suas produções são contabilizadas.

Cebolas, cenouras e morangos são representados por textos (@, %, \*, respectivamente), repolhos, por retângulos e abóboras por circunferências. Pragas, defensivos e adubos são como “gotículas” circulares despejados numa certa região. Uma praga, ao atingir uma hortalica, danifica-a em uma certa medida, podendo, inclusive, matá-la. Um defensivo repara, também em certa medida, repara a hortalica. O adubo faz crescer a hortalica.

A colheita é feita por uma colheitadeira. A colheitadeira é um retângulo que representa sua plataforma de colheita. Ou seja, ela colhe tudo o que estiver **inteiramente** contido dentro deste retângulo. A colheitadeira se desloca em uma direção em passos de mesma magnitude daquela direção. A Ilustração 1 mostra um terreno com várias hortaliças e uma colheitadeira se deslocando para direita em 4 passos. Note que no primeiro passo a colheitadeira está em sua posição original e colhe alguns morangos. No segundo, ela desloca-se para direita na medida de sua largura e colhe alguns morangos, algumas cebolas e uma abóbora e, assim, sucessivamente.

Finalmente, as hortaliças de uma região podem espalhar suas sementes para uma região próxima de acordo com um certo fator multiplicativo. Por exemplo, se o fator multiplicativo for 2.0, cada morango gerará 2 novos morangos; se for 0.5, são necessários 2 morangos para que um novo morango seja gerado.

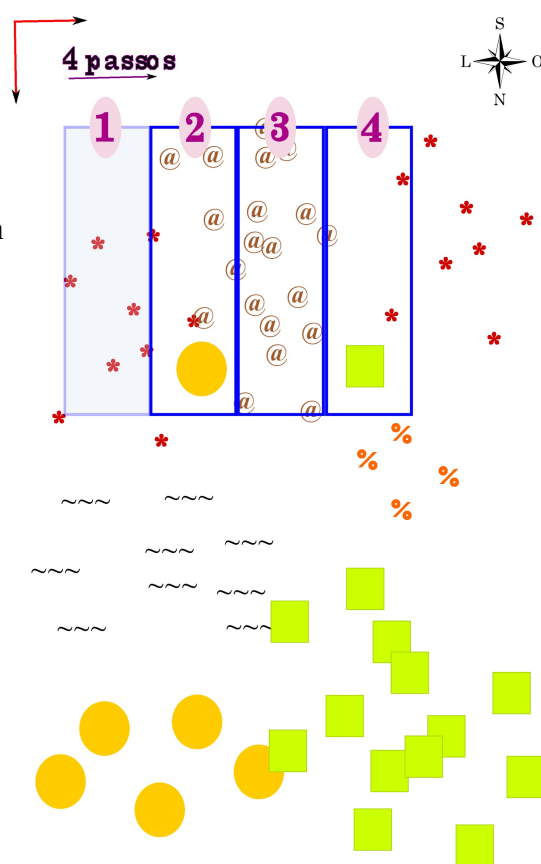


Ilustração 1: Colheita

<sup>1</sup> Veja também a descrição geral para todos os trabalhos.

## A Entrada

A entrada é constituída, basicamente, um conjunto de formas geométricas básicas (retângulos, círculos, etc) dispostos numa região do plano cartesiano .

Considere a Ilustração 2. Cada forma geométrica é definida por uma coordenada âncora (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<sup>2</sup> 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.

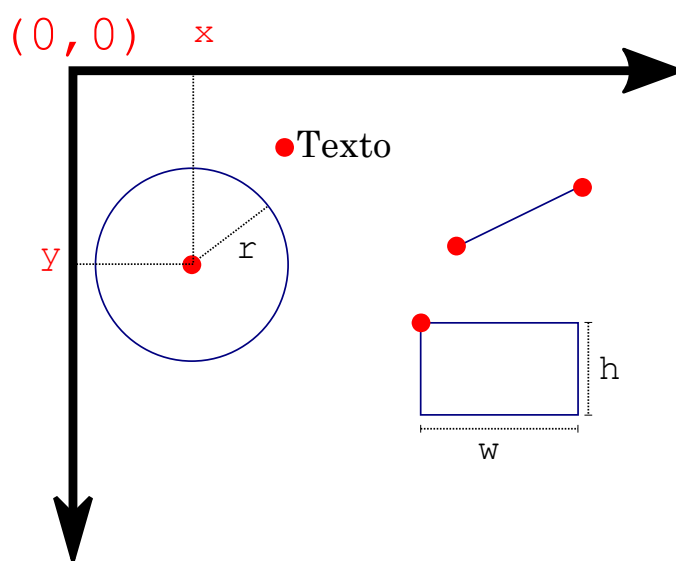


Ilustração 2: Formas no plano

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.<sup>3</sup>

comando	parâmetros	descrição
<b>c</b>	$i \ x \ y \ r \ corb \ corp$	<i>desenhar círculo. corb é a cor da borda e corp é a cor do preenchimento</i>
<b>r</b>	$i \ x \ y \ w \ h \ corb \ corp$	<i>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</i>

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

<sup>3</sup> <http://www.december.com/html/spec/colorsvg.html>.

<https://www.w3.org/Graphics/SVG/IG/resources/svgprimer.html>

<b>l</b>	i x1 y1 x2 y2 cor	Desenhar linha com extremidades nos pontos (x1,y1) e (x2,y2), com a cor especificada.
<b>ts</b>	fFamily fWeight fSize	Muda o estilo dos textos (comando t) subsequentes. font family: sans (sans-serif), serif, cursive; font weight ( n: normal, b: bold, b+: bolder, l:   lighter)
<b>t</b>	i x y corb corp a txto	desenha o texto txto nas coordenadas ( <b>x,y</b> ) e com a cores indicadas. corb é a cor da borda e corp é a cor do preenchimento. O parâmetro <b>a</b> determina a posição da âncora do texto: <b>i</b> , no início; <b>m</b> , no meio, <b>f</b> , no fim. O texto txto é o último parâmetro do comando. Pode incluir espaços em branco e se estende até o final da linha.
<b>comandos .geo</b>		

O arquivo .qry pode conter as seguintes consultas:

comando	parâmetros	descrição
<b>cl</b>	i	Informa que o retângulo de código i é uma colheitadeira.
<b>hvt</b>	i p ( n   s   l   o )	Colheitadeira de código i move-se p passos em uma dada direção (n: norte, s: sul, l: leste, o: oeste). Elementos colhidos são contabilizados e removidos do terreno. TXT: reportar os atributos da figura i, a posição original e a posição final. Reportar os atributos dos elementos colhidos e a contabilidade parcial (i.e., dessa colheita) SVG: desenhar um retângulo de bordas vermelhas pontilhadas indicando a região colhida
<b>mv</b>	i dx dy	Move figura/texto de identificador i em dx (no eixo x) e dy (no eixo y). dx e dy podem ser negativos TXT: reportar os atributos da figura i, a posição original e a posição final.

comando	parâmetros	descrição
<b>ct</b>	x y w h r	<p>Pragas de raio <math>r</math> são uniformemente distribuídas na região retangular <math>(x,y,w,h)</math> danificando cumulativamente as hortaliças atingidas na mesma proporção da área da praga. Caso a hortaliça tenha mais de 75% de sua área atingida, ela é eliminada.</p> <p>TXT: reportar atributos das hortaliças atingidas (não esquecer grau de dano), assinalando as que foram eliminadas.</p> <p>SVG: colocar um retângulo de bordas vermelhas pontilhadas na região <math>(x,y,w,h)</math>, um círculo de <b>borda vermelha</b> raio <math>r</math> em <math>(x,y)</math>. Marcar com um x vermelho a posição da figura eliminada</p>
<b>cr</b>	x y w h r	<p>Similar a ct. Cura as hortaliças na mesma proporção das “gotículas” de raio <math>r</math> do defensivo.</p> <p>TXT: reportar atributos das hortaliças atingidas (não esquecer grau de dano)</p> <p>SVG: colocar um retângulo de bordas vermelhas pontilhadas na região <math>(x,y,w,h)</math>, um círculo de <b>borda amarela</b> e raio <math>r</math> em <math>(x,y)</math>.</p>
<b>ad</b>	x y w h r	<p>Similar a ct. Sempre que a área integral da hortaliça for atingida pelo adubo, a produtividade<sup>4</sup> da hortaliça é incrementada em 10%.</p> <p>TXT: reportar atributos das hortaliças atingidas.</p> <p>SVG: colocar um retângulo de bordas vermelhas pontilhadas na região <math>(x,y,w,h)</math>, um círculo de <b>borda verde</b> e raio <math>r</math> em <math>(x,y)</math>.</p>
<b>st</b>	x y w h f dx dy j	<p>Dispersa as sementes das hortaliças que estão dentro da região <math>(x,y,w,h)</math>, a um deslocamento de <math>dx</math>, <math>dy</math>, com fator <math>f</math>, com identificadores a partir de <math>j</math>.</p> <p>TXT: reportar id e atributos dos elementos atingidos; id e atributos dos clones; reportar os atributos dos clones.</p> <p>SVG: colocar um retângulo de bordas vermelhas pontilhadas na região <math>(x,y,w,h)</math> e outro semelhante na região deslocada. Colocar um pequeno círculo com preenchimento vermelho em <math>(x,y)</math></p>
<b>d?</b>	i	<p>Dados da figura de identificador <math>i</math>.</p> <p>TXT: reportar todos os atributos da figura de código <math>i</math>.</p>

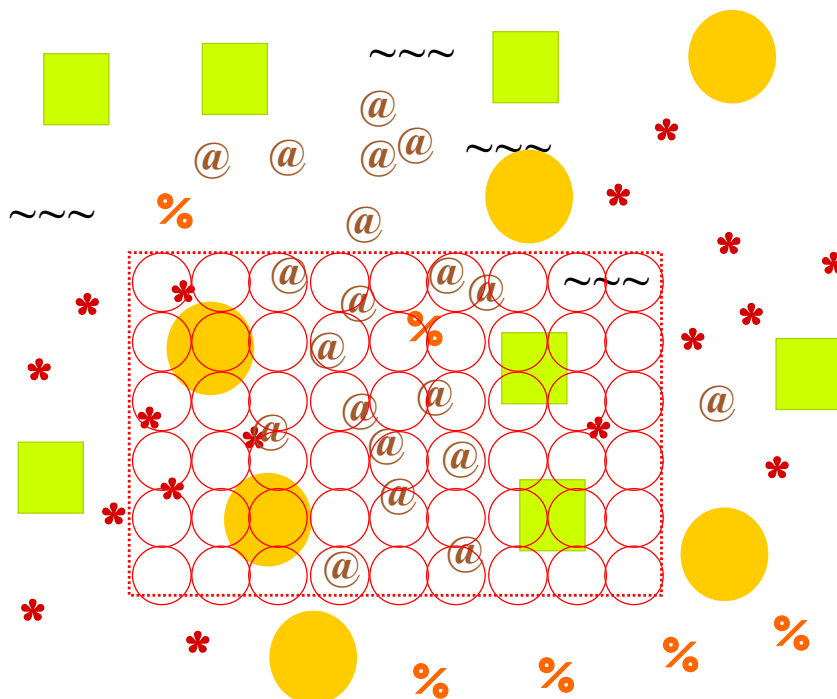
4    Peso da hortaliça. Ver tabela.

comando	parâmetros	descrição
c?		<i>TXT: reporta os atributos de todas as colheiteiras.</i>
Comandos .gry		

## Pragas, Defensivos, Adubos e Sementes

A Ilustração 3 representa qualquer um dos comandos (ct, cr e ad). A figura mostra a região onde as “gotículas” são aplicadas (seja uma praga, um defensivo, ou um adubo) e as “gotículas” distribuídas uniformemente dentro desta região.

A gotícula só produz efeito sobre a hortaliça caso ela esteja completamente interna à hortaliça (aquelas representadas por retângulos e círculos) ou que as contenham inteiramente (no caso daquelas representadas por textos e linhas)



*Ilustração 3: Representação de comando ct, ad, ou cr*

Note que os efeitos destes comandos são cumulativos e que os efeitos são **proporcionais às áreas da gotícula e da hortaliça atingidas**. No caso de textos e linhas, a proporção é **fixa em 10%**.

A Ilustração 4 exemplifica o comando st. Note as “sementes” existentes na região retangular mais à esquerda germinaram (região retangular mais à direita) numa proporção de 1:2.

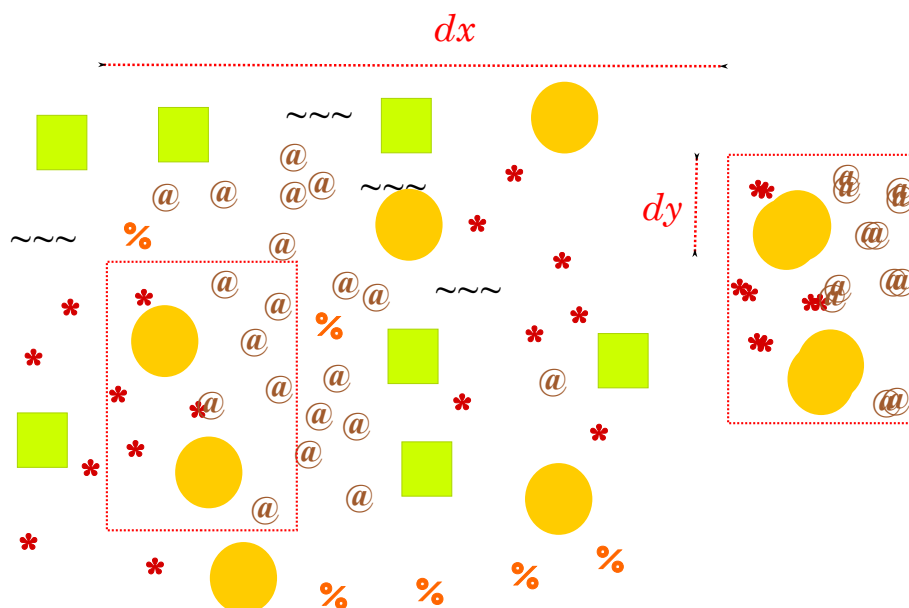


Ilustração 4: st x y dx dy 2.0 j

## Contabilidade da Colheita

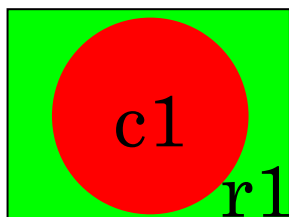
A contabilidade da colheita resume-se em calcular o peso total de cada hortaliça colhida. A tabela abaixo relaciona o peso inicial de cada uma. Este peso pode ser modificado pela ocorrência de pragas, pela aplicação de defensivos e de adubos.

Peso de cada produto:

Produto	Peso
abóbora	2kg
morango	20g
repolho	1kg
cebola	200g
cenoura	70g
mato(linha)	10g por unidade de comprimento
mato(texto)	15g
Peso das hortaliças	

Para contabilidade, primeiro calcular o peso corrente da hortaliça, resultante da adubação. Note porcentagem da “área adubada” pode ser maior que 100%. Lembre-se que, a cada 100%, deve ser acrescentada 10% do peso inicial. A seguir, descontar a perda pela ocorrência de pragas (não curadas). Por exemplo, considere a Ilustração 5. Suponha que a circunferência c1 possua centro em (28.00 , 24.00) e raio 13.00. Ainda suponha que o retângulo r1 possua âncora em (10.00,

10.00), largura 37.60 e altura de 28.20). Assim, a área do retângulo é o dobro da área da circunferência.



$$\text{area}(r1) = 2 * \text{area}(c1)$$

Ilustração 5: Hortaliça e "gotícula"

Considere, ainda os arquivos geo e qry mostrados na tabela abaixo:

Arquivo .geo							
r	5	10.00	10.00	37.6	28.2	black green	1
Arquivo .qry							
ct	15.00	10.00	50.00	60.00	13.00		1
cr	15.00	10.00	50.00	60.00	13.00		2
ct	15.00	10.00	50.00	60.00	13.00		3
cr	15.00	10.00	50.00	60.00	13.00		4
ct	15.00	10.00	50.00	60.00	13.00		5
ad	15.00	10.00	50.00	60.00	13.00		6
ad	15.00	10.00	50.00	60.00	13.00		7
ad	15.00	10.00	50.00	60.00	13.00		8
ad	15.00	10.00	50.00	60.00	13.00		9
ad	15.00	10.00	50.00	60.00	13.00		10

Os comandos ct, cr e ad do arquivo .qry podem produzir “gotículas” sobre o retângulo 5 como mostrado na Ilustração 5. Note que a proporção área(gotícula)/área(r) é 0.5. A execução das linhas 1-4 alternam ocorrência de praga e uso de defensivo nas mesmas proporções, o segundo curando o primeiro, de forma que nenhum dano ocorre. Na linha 5, ocorre praga, **comprometendo** a hortaliça **em 50%**. As linhas 6-10 adubam a mesma hortaliça, também, na proporção de 0.5, totalizando 250% de área adubada. Portanto, a hortaliça terá um **acréscimo de 20%** do peso inicial.<sup>5</sup> Assim, o retângulo 1, é um repolho, portanto com **peso inicial de 1kg**, foi acrescido, por adubação de **200g** e, finalmente, perdeu metade de seu peso por ocorrência de pragas. O **peso final** do repolho é **600g**.

## A Saída

Ao final do processamento do arquivo .qry é produzido o svg representando a situação final do “banco de dados”, com as anotações descritas nas especificações.

Reportar no txt a contabilidade total das hortaliças colhidas e a contabilidade das hortaliças não colhidas (por espécie).

Também, deve ser produzido um arquivo .dot que “desenha” o estado final da árvore.<sup>6</sup>

<sup>5</sup> Os últimos 50% de área adubada não produziram nenhum ganho de peso.

<sup>6</sup> Consultar: <https://graphviz.org/>

## IMPLEMENTAÇÃO

O “banco de dados” **deve** ser armazenado numa árvore radial (**deve** ser conforme .h disponibilizado no AVA). A Ilustração 6 e a Ilustração 7 exemplificam uma árvore com seis setores.<sup>7</sup> Note, na primeira os pontos que estão dentro de cada setor referente ao ponto central (estão em tonalidades semelhantes). A primeira ilustração também exemplifica uma forma de determinar o setor no qual um dado ponto  $p$  é interno. A segunda ilustração mostra os pontos no plano e o desenho da árvore resultante da inserção desses pontos na ordem indicada pelos números.

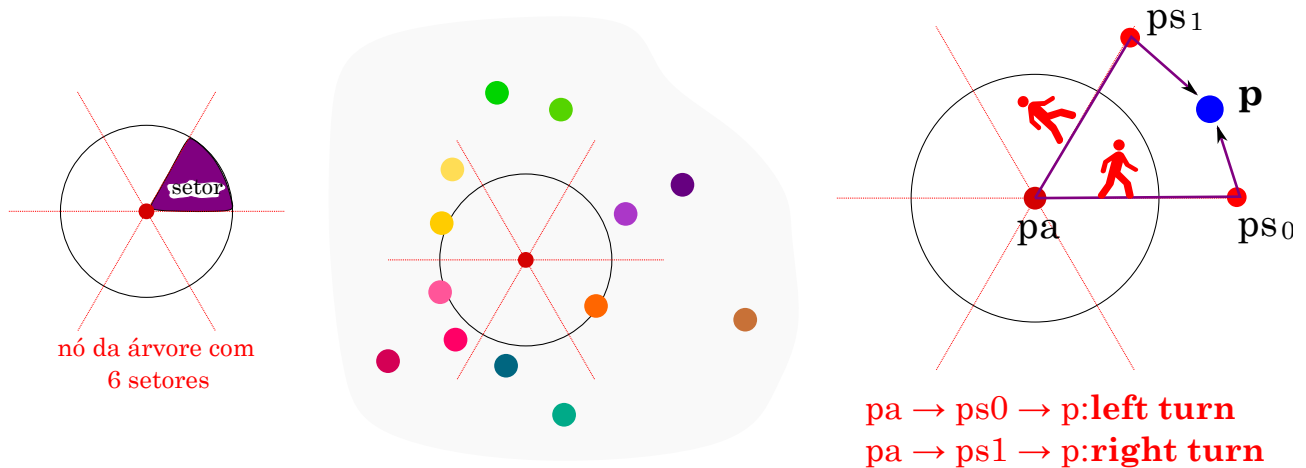


Ilustração 6: Nó da árvore com 6 setores

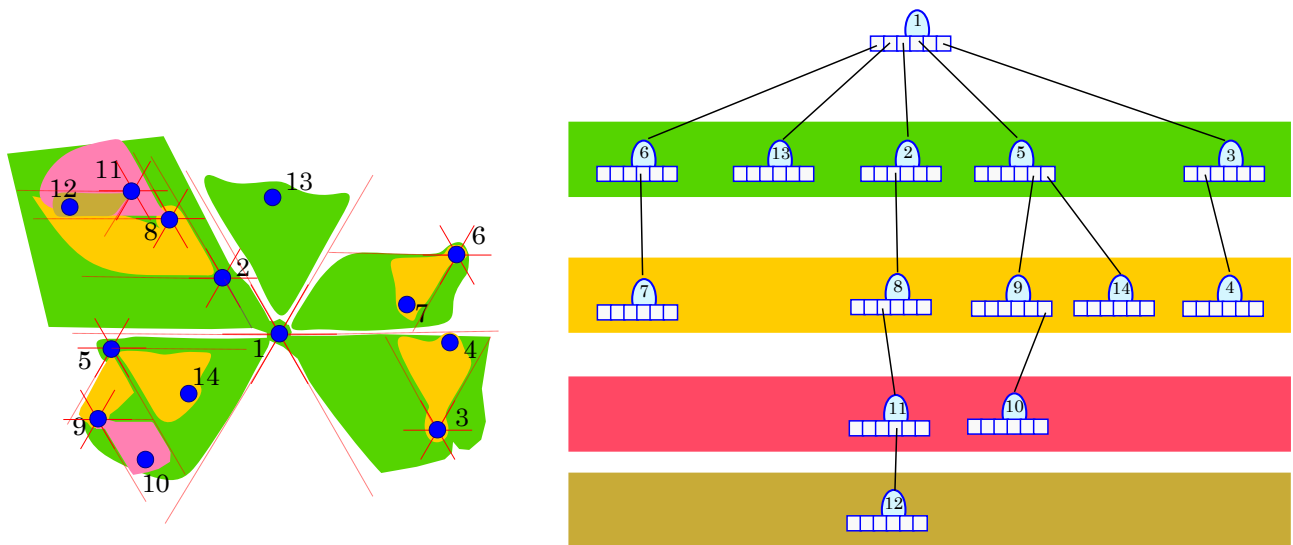


Ilustração 7: Exemplo de uma árvore. O número dos nós corresponde à ordem de inserção.

As operações de busca por região **devem** “podar” a busca. A Ilustração 8 mostra um exemplo. Note que a região de busca está representada pelo retângulo pontilhado vermelho. Em relação ao nó 1 (raiz), a região de busca apenas intercepta os setores 0 e 5. Portanto, a busca deve descer apenas nas sub-árvores relativas a tais setores. Assim, a busca **não** desce nas sub-árvores dos outros setores (ou seja, tais sub-árvores são “podadas”).

<sup>7</sup> O número de setores é passado na invocação do programa (veja parâmetro -ns).



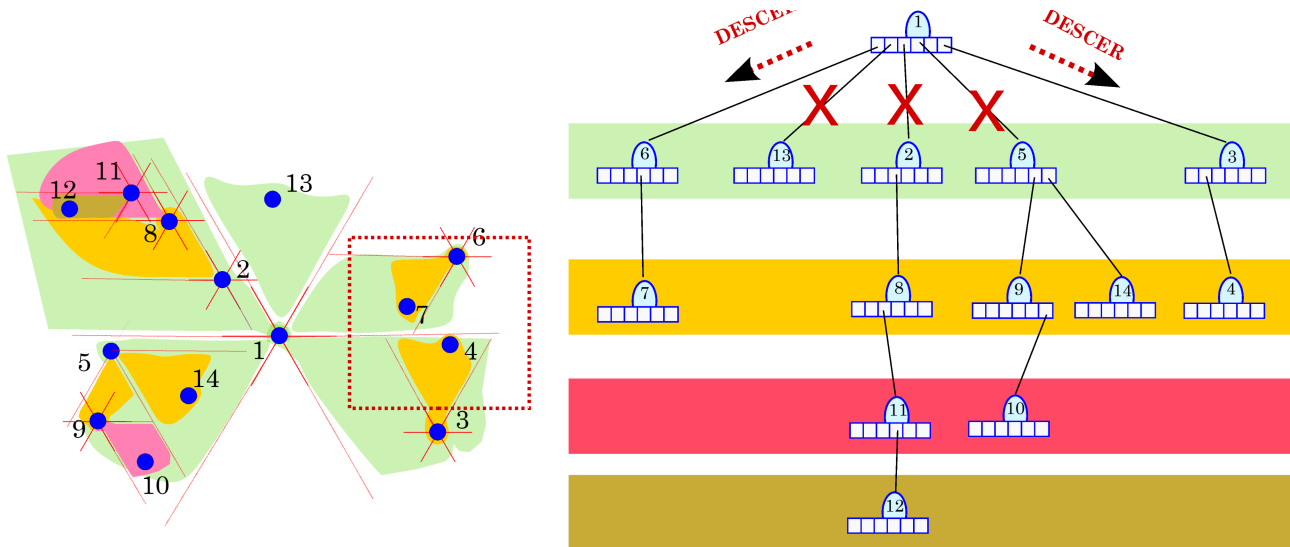


Ilustração 8: Exemplo de poda.

Sempre que a árvore for reorganizadas, os valores devem ser re-inseridos numa ordem que produza uma árvore razoavelmente bem distribuída. Para que isto ocorra, considere o centro do menor retângulo que envolve as âncoras dos dados **não** removidos. Os elementos mais próximos deste centro devem ser re-inseridos antes dos mais distantes. Usar o algoritmo quicksort provido pela biblioteca padrão da linguagem C. A Ilustração 9 mostra um conjunto de pontos armazenados numa árvore radial que está prestes a ser reorganizada. Os pontos azuis são pontos válidos que deverão ser re-inseridos. Os cinzas foram marcados como removidos e, portanto, não serão re-inseridos. O ponto  $p$  (laranja) é um ponto fictício que está no centro do retângulo que envolve os pontos válidos. Note que  $p_i$  será re-inserido antes de  $p_j$ , pois a distância euclidiana de  $p_i$  para o ponto central  $p$  é menor que a distância de  $p_j$  até este ponto central.

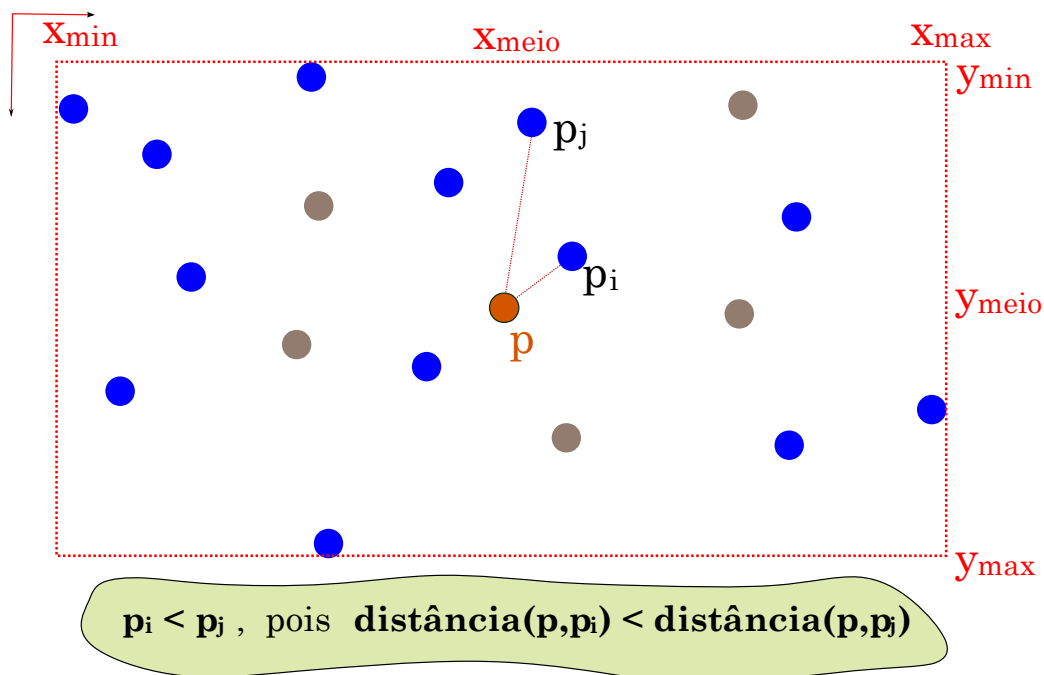


Ilustração 9: Pontos azuis (válidos), pontos cinzas (marcados como removidos)

Além disso, não esquecer:

- **Usar** o modelo de makefile disponibilizado no AVA da disciplina.
- 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”).
- Testar o zip antes de enviá-lo.

## 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.

A nota é proporcional ao número de testes corretos, aplicados os descontos abaixo:

Critério	Desconto
Escrever struct em arquivo .h	2.5
Modularização Pobre: .h mal projetado, mal documentado	até 1.5
Não implementado conforme especificado <i>Muito importante: poda nas buscas, reorganização após muitas delições, número de setores definidos na criação da árvore (-ns), desenho da árvore, fator de degradação variável (-fd)</i>	tipicamente até 3.0, mas em casos graves, o desconto pode ser total
Procedimentos extensos e/ou complicados	até 1.0
Não usar o makefile provido	Em geral, nenhum desconto. Mas, se ocorrer problema de compilação ou execução que poderia ter sido evitado pelo uso do modelo do makefile provido, a consequência pode ser grave.
Erro de compilação	Nenhum teste será executado. Portanto, nota Zero, especialmente se for causado por negligência do aluno.
Não usar o quicksort	até 1.5
Não reorganizar a árvore como indicado	até 5.0

## 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 <i>path</i>	S	Diretório-base de entrada ( <b>BED</b> )
-f <i>arq.geo</i>	N	Arquivo com a descrição da cidade. Este arquivo deve estar sob o diretório <b>BED</b> .
-o <i>path</i>	N	Diretório-base de saída ( <b>BSD</b> )
-q <i>arqcons.qry</i>	S	Arquivo com consultas. Este arquivo deve estar sob o diretório <b>BED</b> .
-ns <i>i</i>	N	Número de setores em cada nó da árvore
-fd <i>p</i>	N	Fator de degradação (em porcentagem. exemplo: 75 => 0.75)

## RESUMO DOS ARQUIVOS PRODUZIDOS

-f	-q	comando com sufixo	arquivos
<i>arq.geo</i>			arq.svg arq.dot
<i>arq.geo</i>	<i>arqcons.qry</i>		arq.svg arq-arqcons.svg arq-arqcons.txt arq-arqcons.dot
<i>arq.geo</i>	<i>arqcons.qry</i>	<i>sufx</i>	arq.svg arq-arqcons.svg arq-arqcons.txt arq-arqcons.dot arq-arqcons-sufx.[svg txt] <sup>8</sup>

### ATENÇÃO:

\* os fontes devem ser compilados com a opção `-fstack-protector-all`.

\* adotamos o padrão C99. Usar a opção `-std=c99`.

### DÚVIDAS:

<sup>8</sup> Podem ser produzidos os respectivos arquivos .svg e/ou .txt, dependendo da especificação do comando.