


		5	4	-1	-1	1		automax									
Aluno		tests Norm	test esfera	prob compil	prob exec	rel		Soma AUTO	Soma Relat.	atras o	Total	VO(us)					
Bellargus Silva da Costa	Alexandre	1	1			1	i	9.0	1.0		10.0	724					
Diogo Firmino dos Santos	Andre	1	1			1	i	9.0	1.0		10.0	980					
Mesquita Rios	Andre	1	1			1		9.0	1.0		10.0	1200					
Gustavo Silveira Dantas	Antonio	1	1			1	i	9.0	1.0		10.0	1200					
Daigo Yamamoto	Bruno	1	1			1		9.0	1.0		10.0	1530					
Marçal Guimarães	Cau	1	1			1		9.0	1.0		10.0	920					
Araujo Cavassani	Daniel	1	1			1	i	9.0	0.3	1.0	8.3	878					
Derlian Carvalho Brito	Denys	1	1			1		9.0	1.0		10.0	900	Node::construtor deveria fazer o q createNode faz				
Vasconcelos Bastos Neto	Deodato	1	1			1		9.0	1.0		10.0	920					
Reis Spirandelli	Felipe	1	1			1		9.0	1.0		10.0	829	k+logn != log(k+n)				
da Silva Maia Filho	Francisco Gal	1	1			0.8		9.0	0.8		9.8	830	olha com alguém como faz o fit log				
Telles Missailidis	Gabriel	1	1			1		9.0	1.0		10.0	820					
Vicente Batista	Giuseppe	1	1			1	b	9.0	1.0		10.0	3300					
Alt Chagas Merklein	Guilherme	1	1			1		9.0	1.0		10.0	1600	classe RBTREE deveria fazer a busca				
Padua Beato	Gustavo	1	1			1		9.0	1.0		10.0	815					
Lucas Rocha Rolim	Joao	1	1			1		9.0	1.0		10.0	1280	pode usar a queue da STL				
Yoshiro Nishiura	Juliano	1	1			1		9.0	1.0		10.0	2600	zip o diretorio, não os arquivos dentro dele!				
Georges Balech	Kalil	1	1			1	i	9.0	1.0		10.0	1111	Kalil.zip não src.zip. Use o script				
Schneider Friedrich	Leonardo	1	1			1	i	9.0	1.0		10.0	800					
Nogueira Costa	Lucas	1	1			1		9.0	1.0		10.0	940					
Silva Lima	Lucas	1	1			1	i	9.0	1.0		10.0	750	interface pseudo-iterator c/ ponteiros. rapido				
Moura Zoppi Maia	Lucca	1	1			1	o	9.0	1.0		10.0	1540					
Rangel Haddad	Lucca	1	1			1		9.0	1.0		10.0	900					
Versiani e Silva	Marcel	1	1			1	i	9.0	1.0		10.0	790					

		5	4	-1	-1	1		automax									
Aluno		tests Norm	test esfera	prob compil	prob exec	rel		Soma AUTO	Soma Relat.	atras o	Total	VO(us)					
Melo de Oliveira	Marcelo	1	1			1		9.0	1.0		10.0	1370					
Moreira Goncalves Feltrin	Moises	1	1			1		9.0	1.0		10.0	950					
da Silva Antes	Nikollas	1	1			1		9.0	1.0		10.0	3200					
Daiki Ishiyi	Othon	1	1			1		9.0	1.0		10.0	1800					
Henrique Ferron Kim	Pedro	1	1			0		9.0	0.0		9.0	900	pode ter 2 construtores com args diferentes				
Igor Dourado Borges	Pedro	1	1			1		9.0	1.0		10.0	1340	cuidado, ha atrib res. mesmo nome do param local.				
Camargo	Rafael	1	1			1		9.0	1.0		10.0	1000					
Hoffmann Giannico	Rafael	1	1			1		9.0	1.0		10.0	1500					
Otero Litran Sato	Rafael	1	1			1		9.0	1.0		10.0	1000	tive q acrescentar #include<vector> em RbTree.cpp. cuidado				
Nunes da Silva	Samir	1	1			1		9.0	1.0		10.0	880					
Athanassios Marathaoan C	Stelios	1	1			0.2		9.0	0.2		9.2	910	nao coloque diretorio no #include "../src/File.h" meu script não executa no src				
Jose de Menezes Pereira	Vinicius	1	1			0.2		9.0	0.2	1.0	8.2	910					
Mendes de Almeida	Yuri	1	1			0.2		9.0	0.2		9.2	1000					
	#corr	37	37	0	0	37				2							
OU COLOCA O PDF DENTRO DO DIRETORIO SOURCE, OU FORA MAS CHAMADO SEUNOME.PDF. ARQUIVO RELATORIO.PDF FORA DO SOURCE DIFICULTA																	
sobre as implementacoes diferentes: eh interessante, mas parecem confirmar que a forma convencional do livro-texto eh o melhor trade-off espaco-tempo pros casos mais comuns																	
iterator é mais q retornar ponteiros e encontrar successor. isso tudo fica encapsulado em uma classe Iterator. mas valeu o interesse em aprender																	
a)																	
grafico sem nenhum label ou legenda, nao sei o que esta la, tive que entender olhando o arquivo texto																	

		5	4	-1	-1	1	automax										
Aluno		tests Norm	test esfera	prob compil	prob exec	rel	Soma AUTO	Soma Relat.	atras o	Total	VO(us)						
b)																	
ficara cada vez mais dificil escrever todo o codigo em um .h. precisa aprender a separar as classes em .cpp e .h convenientes																	
nao eh tao importante pq os projetos ainda sao pequenos, mas em um projeto maior atrapalha.																	
dei a dica das clausulas de barreira #ifdef no inicio do .h. Codigo em .h demora mais pra compilar tambem pq um arquivo #inclui o outro																	
c)																	
seria melhor modelar direito as classes com menos funcoes C que enxergam demais das classes. nao eh dificil, vc aprende mais																	
g)																	
codigo de funcoes longas em .h trarao problemas em projetos maiores: todos os .cpp que incluirem a sua classe precisarao compilar o codigo																	
i)																	
modelagem: struct em C pode ter construtor, nao so classe. mesmo que node seja struct, construtor evita varias linhas inicializando campo por campo																	
mais facil errar algo com malloc ou new sem argumentos do que com um construtor para a struct. A sintaxe eh mais segura pro humano programador																	
j)																	
modelagem: a classe arvore deveria fazer as operacoes de arvore. ficou uma operacao de arvore na classe que so usa uma arvore, nao eh uma arvore.																	
k)																	
usar deque gastou + tempo, quase 2x. Mas, nao precisa de ponteiro p/ o pai em cada no: trade-off espaco x tempo, fair enough (embora priorizar tempo seja + comum nesse caso)																	
l)																	
interessante manter ponteiro para o prox e prev elemento em cada no.																	
ficou + rapido o VoidSphere, ass pouco, uns 20%. nao parece compensar o gasto O(n) de espaco adicional (no teste minusculo que fizemos).																	
m)																	
usou arvore implementada em vetor, percorrendo indices pra fazer busca e trocas pra balancear.																	
o gerenciamento de memoria do std::vector ferrou com o seu tempo, ele desaloca e aloca o dobro toda vez que o tamanho aumenta de potencia de 2																	
entao vc sem perceber desalocou varias vezes o seu vetor. ficou parecido em tempo com a stl, mas quem usou arvore convencional ficou mais rapido que a STL																	
n)																	

		5	4	-1	-1	1		automax									
Aluno		tests Norm	test esfera	prob compil	prob exec	rel		Soma AUTO	Soma Relat.	atras o	Total	VO(us)					
eh esperado que AVL seja mais lenta pra insercao, eh muito rebalanceamento, quem implementou RBtree ficou mais rapido que a STL, vc ficou igual.																	
E espero que tenha aprendido alguma coisa com o std::lterator - mesmo que tenha visto uma receita de bolo, eh assim que vai pegando experiencia																	
o)																	
mais lento pq copiou vector inteiro na resposta. poderia passar direto o std::vector res com parametro & no seu metodo find, e escrever os elementos no vetor uma vez só.																	