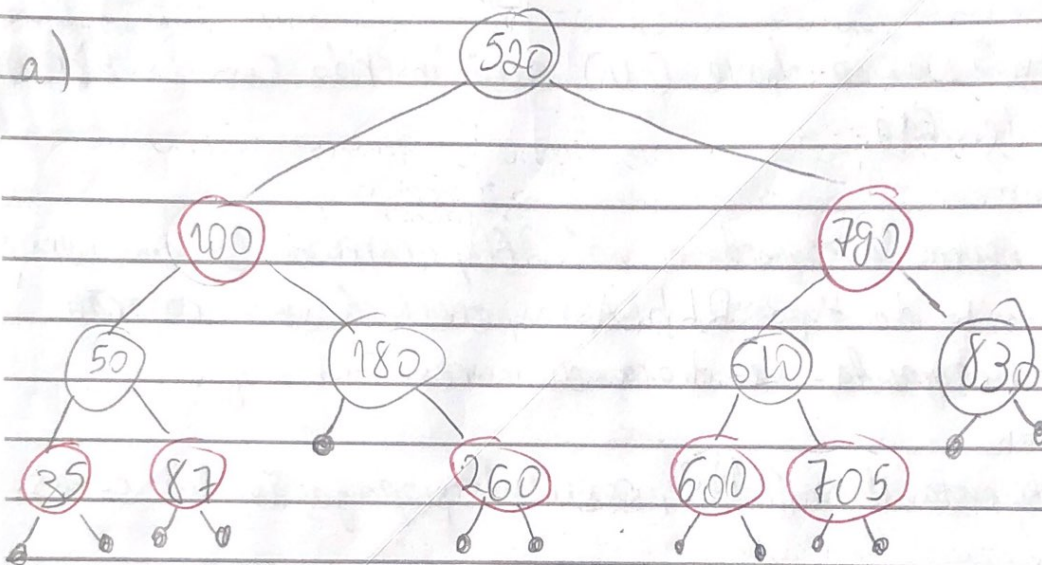


P2 - INF1010

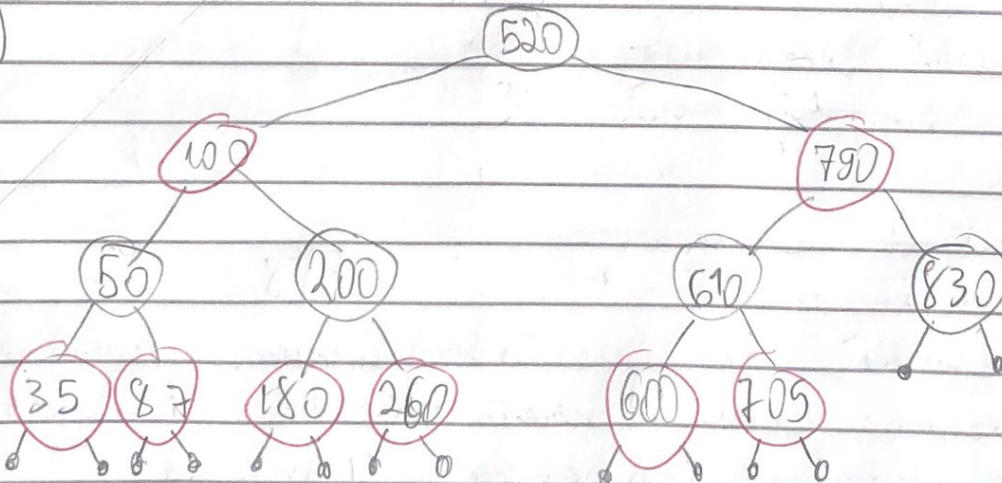
João Pedro Khair Cunha - 1910626

Questão 1:

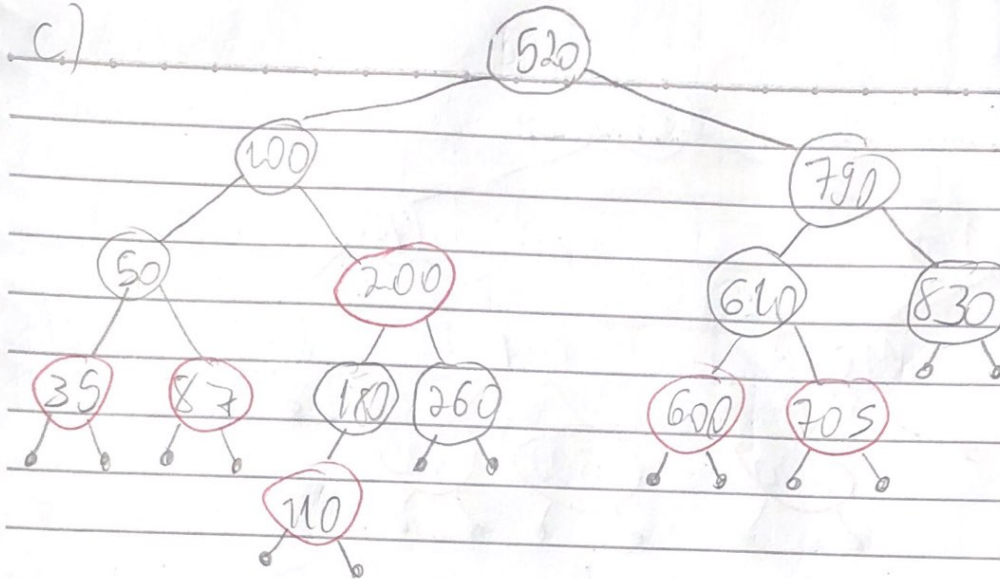
a)



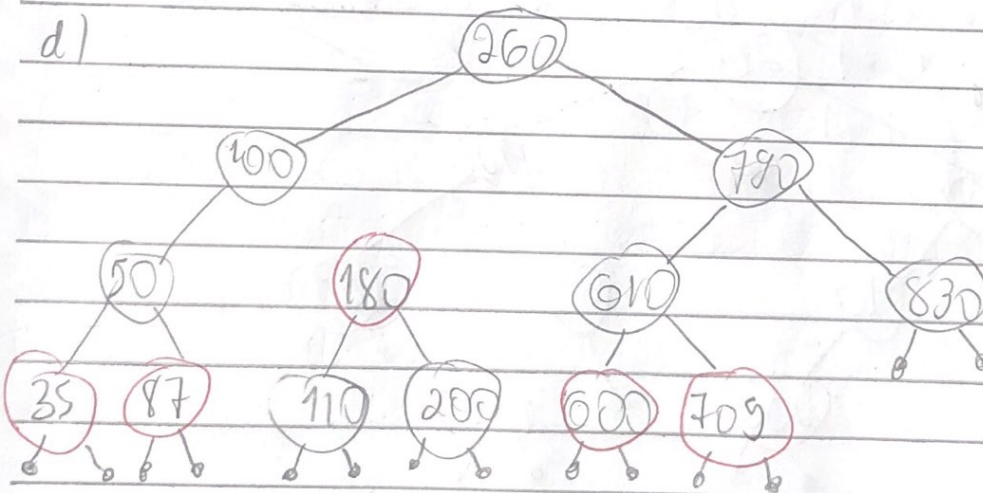
b)



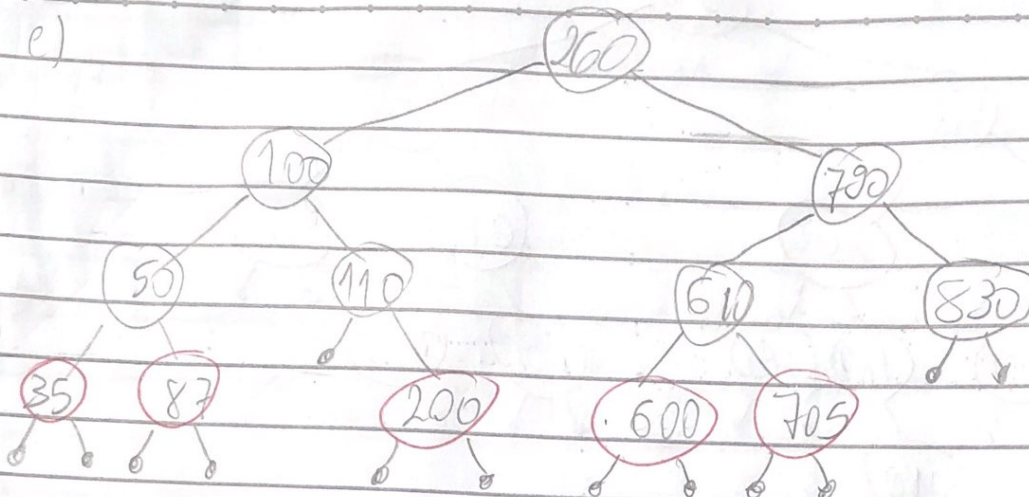
c)



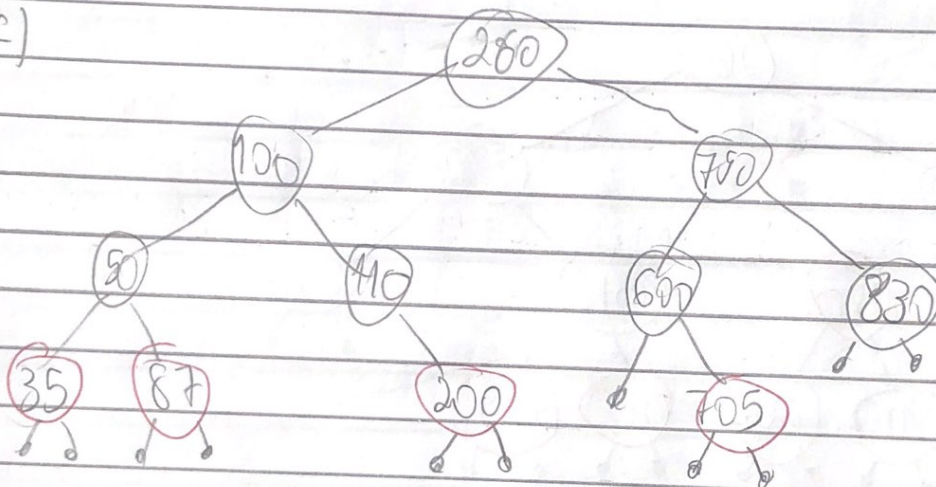
d)



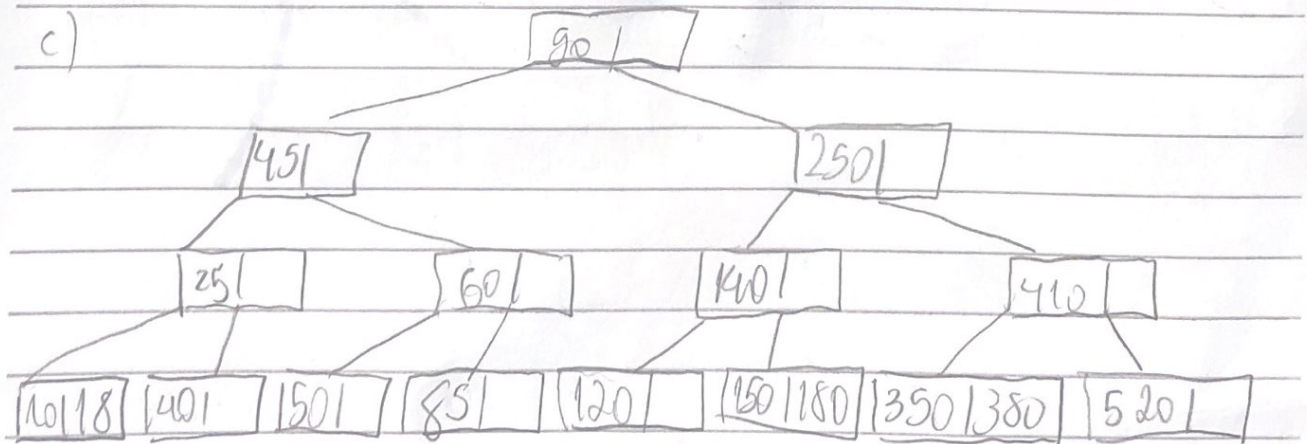
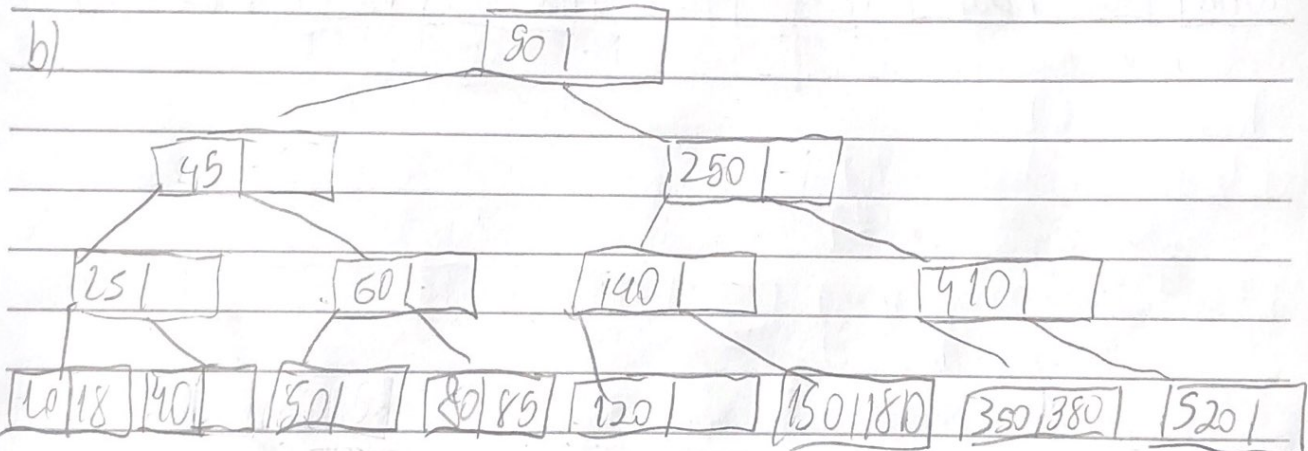
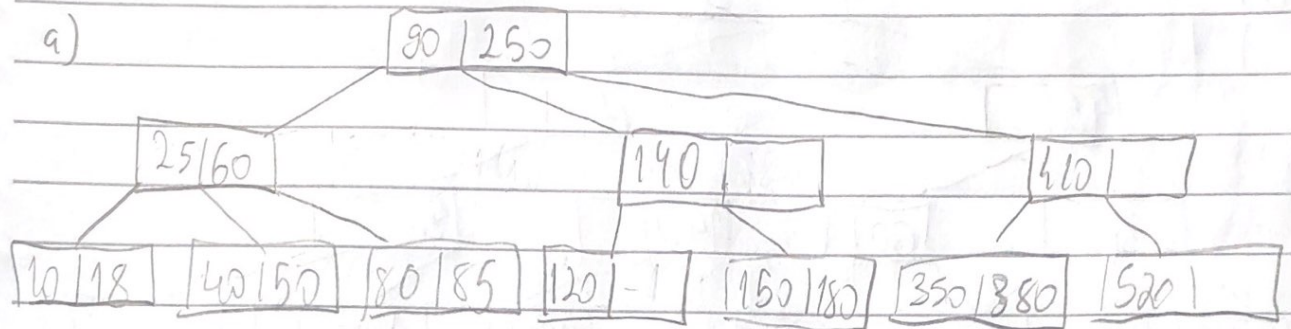
e)

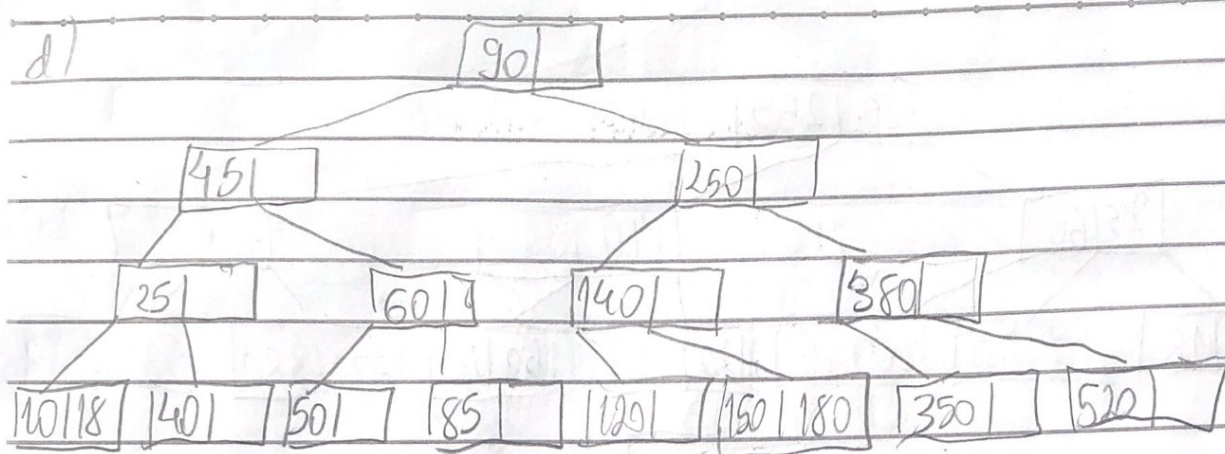


f)



Questão 2:





Questão 3

a) $h(9) = 9, K=0$

0	1	2	3	4	5	6	7	8	9	10
									9	

$h(10) = 10, K=0$

									9	10
--	--	--	--	--	--	--	--	--	---	----

$h(17) = 6, K=0$

						17			9	10
--	--	--	--	--	--	----	--	--	---	----

$h(3) = 3, K=0$

			3			17			9	10
--	--	--	---	--	--	----	--	--	---	----

$h(2) = 2, K=0$

		2	3			17			9	10
--	--	---	---	--	--	----	--	--	---	----

$h(13) = 4, K=2$ (conflicto)

		2	3	13				17		9	10
--	--	---	---	----	--	--	--	----	--	---	----

$h(15) = 4, K=0 \rightarrow$ conflito com 13, que pula uma casa

		2	3	15	13	17			9	10
--	--	---	---	----	----	----	--	--	---	----

$h(25) = 7, K=4$ (conflictos)

		2	3	15	13	17	25		9	10
--	--	---	---	----	----	----	----	--	---	----

$h(14) = 8, K=5$ (conflictos)

		2	3	15	13	17	25	14	9	10
--	--	---	---	----	----	----	----	----	---	----

3.6) (i) Ao remover a chave 3, temos:

1	2	15	13	17	25	14	9	10
---	---	----	----	----	----	----	---	----

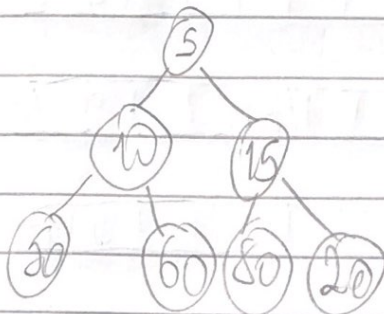
(ii) Ao inserir a chave 4 $\rightarrow h(4)=4, K=0$. Ocorre concorrência até que $h(4)=0, K=7$.

4	2	15	13	17	25	14	9	10
---	---	----	----	----	----	----	---	----

Questão 4:

a) Arquivo .c enviado com este pdf.

b)



c) O heapsort é um algoritmo que ordena o vetor do heap do final para o início. Inicialmente, ele constrói um heap bottom-up para, então, "criar" o vetor ordenado FORA do heap \rightarrow info. Isso é feito colocando a raiz (maior elemento) no final de heap \rightarrow info, para então diminuir em 1 unidade o vetor. Assim, chama-se novamente a função heapify a partir da raiz e recomeça o processo até que heap \rightarrow pos seja 1. Então, retomamos o valor inicial de heap \rightarrow pos, obtendo o vetor ordenado

[5 | 10 | 15 | 50 | 60 | 80 | 20] constrói-heap [80 | 60 | 15 | 50 | 10 | 15 | 20]

[50 | 20 | 15 | 15 | 10 | 60 | 80] \leftarrow [60 | 50 | 5 | 20 | 10 | 15 | 80]

[20 | 15 | 5 | 10 | 50 | 60 | 80] \rightarrow [15 | 10 | 5 | 20 | 50 | 60 | 80]

[5 | 10 | 15 | 20 | 50 | 60 | 80] \leftarrow [10 | 5 | 15 | 20 | 50 | 60 | 80]

\rightarrow [5 | 10 | 15 | 20 | 50 | 60 | 80] ORDENADO