Aufgabe 1

(5) 8 3 7 1 4 2 9 6

6 8 3 7 1 4 2 9 (5)

i j

2 8 3 7 1 4 6 9 (5)

i j

2 4 3 7 1 8 6 9 (5)

i j

2 4 3 1 7 8 6 9 (5)

2 4 3 1 | 5 | 8 6 9 7

(2) 4 3 1

1 4 3 (2)

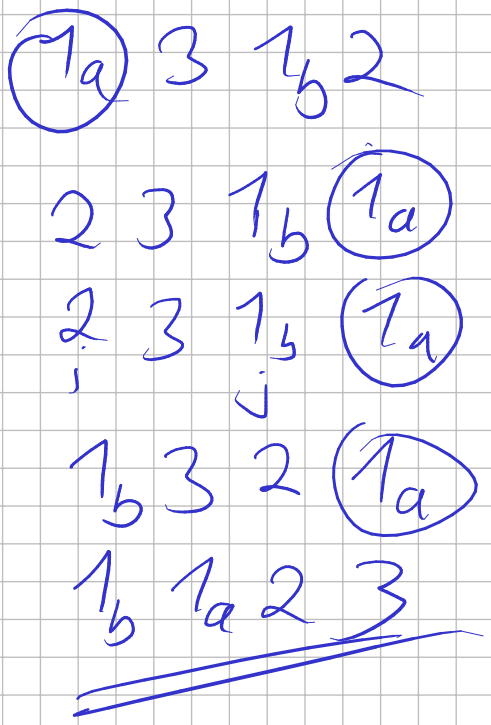
| 1 2 3 4 5 | 8 6 9 7

8 6 9 (7)

i j

6 8 9 (7)

| 1 2 3 4 5 6 7 8 9 |



Aufgabe 2

- Ja, indem man beim suchen des kleinsten Elements, das Array aufteilt und für die Sub arrays das min-Element berechnet. Dies wiederholt man dann rekursiv bis nur noch ein Element übrig ist.
- Nein, weil ein fertiges Element muss sich in dem bereits sortierten Teil einordnen
- Nein, removeMin erwartet den neusten Baum.
- Ja, da man operationen auf den neuen Teilmengen unabhängig durchführen kann
- Ja, da man operationen auf den neuen Teilmengen unabhängig durchführen kann

Aufgab 3:

Wenn wir keine Zeit oder Leistungspakete mehr übrig haben ist die Rückgabe 0. Ist da jetzig Leistungspaket zu aufwendig, probieren wir das nächste. Passt es, probieren wir es mit dem Leistungspaket und ziehen dessen

dauer von der gesamtdauer ab. Außerdem prüfen wir es auch mit der nächsten Leistung später.

2. E [9500, 6500, 13000] M [2,3,5] m = 12 n = 3

	3	2	1	gesamt €
1	0	0	6	27.000
2	0	1	4	29.500
3	0	2	3	26.500
4	0	3	1	24.000
5	0	4	0	26.000
6	1	0	3	26.500
7	1	1	2	27.500
8	1	2	0	27.000
9	2	0	1	30.500

3, 3, 1 = 30.500 €

3) Man kann für jede Iteration eine Liste erstellen, auf die immer, wenn die Zeit abgezogen wird, ein Element gepusht wird. Am ende wird nun nicht die 0, sondern die Liste zurückgegeben.