To find the optimal cutoff and the number of threads, I wrote a loop that changes the size of the cutoff from 1 to 100'000'000 in steps of the previous cutoff times 10.
And the threads I tested the program for 2 … 8192 threads for each cutoff size

Programm snippet:

```
for(int i = 1; i <= 100000000; i = i * 10) {
   for(int j = 2; j <= 8192; j = j * 2) {
      System.out.println("Testing  program with " + i + " cutoff and " + j + " threads");
      taskB(input, i, Workload.Type.HEAVY, j);
   }
}
```

	1	10	100	1000	10000	100000	1000000	10000000	100000000	
2	110	32	47	47	32	62	47	63	62	55,78
4	109	47	31	78	31	63	63	66	47	59,44
8	94	46	47	63	32	62	62	48	63	57,44
16	47	32	31	46	47	63	47	62	62	48,56
32	31	31	47	32	62	46	78	47	63	48,56
64	47	47	31	31	31	63	47	63	62	46,89
128	47	31	32	78	47	62	62	62	63	53,78
256	31	47	47	47	16	47	63	63	47	45,33
512	47	47	31	31	47	63	62	47	62	48,56
1024	78	47	47	47	47	62	63	62	63	57,33
2048	31	31	31	47	62	47	47	63	47	45,11
4096	47	47	31	62	31	63	62	46	62	50,11
8192	31	31	47	47	32	62	47	63	63	47,00
	57,69	39,69	38,46	50,46	39,77	58,85	57,69	58,08	58,92	

100 cutoff and 2048 seems to result in the best average performance for the submitted testsets