

1. We would like to make a specific amount of water when we have k water bottles and only the following operations are allowed:

- Fill up either bottle completely.
- Empty either bottle completely.
- Pour water from one bottle to the other until the poured bottle becomes empty or the other bottle becomes full.

Write sequential and cluster parallel programs that solve this problem using breadth-first search. The program reads: target_amount, bottle1, bottle2, ..., bottlek where bottlek represents the capacity of the bottle. The output is a sequence of bottle status at each step from the initial state to the target state. You can use the master-worker pattern in the cluster parallel program. Feel free to use other approaches if you find them working better.

The sequential and parallel program for this homework are attached with this report. The readme.txt for compilation and execution for each sequential and parallel.

The test case for which time was measured is given as:

Bottle capacity = 3,5,7,13,19,29,31,37,39,41

Goal = 40

The reported time are:

Sequential	Parallel
7809 ms	5354 ms

Based on above observation, the speed up is calculated as:

Speed up = Time for sequential / Time for parallel
~ 1.45

```
pkg2182@tardis:~/Parallel$ java pj2 jar=test.jar WaterJugParallel
Start time=1491530748975
Job 356 launched Thu Apr 06 22:05:49 EDT 2017
Job 356 started Thu Apr 06 22:05:49 EDT 2017
[0, 5, 7, 13, 19, 29, 0, 0, 0, 40]
Finish Time=1491530754329
Job 356 finished Thu Apr 06 22:05:49 EDT 2017 time 0 msec
pkg2182@tardis:~/Parallel$
```

Fig: The time for parallel is calculated as the difference between the start time and the time when the goal is found by some worker node

Note:

1. The sequential code was executed on the local machine.
2. The parallel code was executed on the tardis.
3. The difference in time for the execution was observed for both sequential and parallel version.