Aufgabe 1

a)

T = 1 hour * (washing machine + dryer + flatiron) * 5 wash loads = 1 * 3 * 5 = 15 hour

Because working with serial loop, each wash load need 3 hours. And there are 5 wash loads, so they need 15 hour.

b)

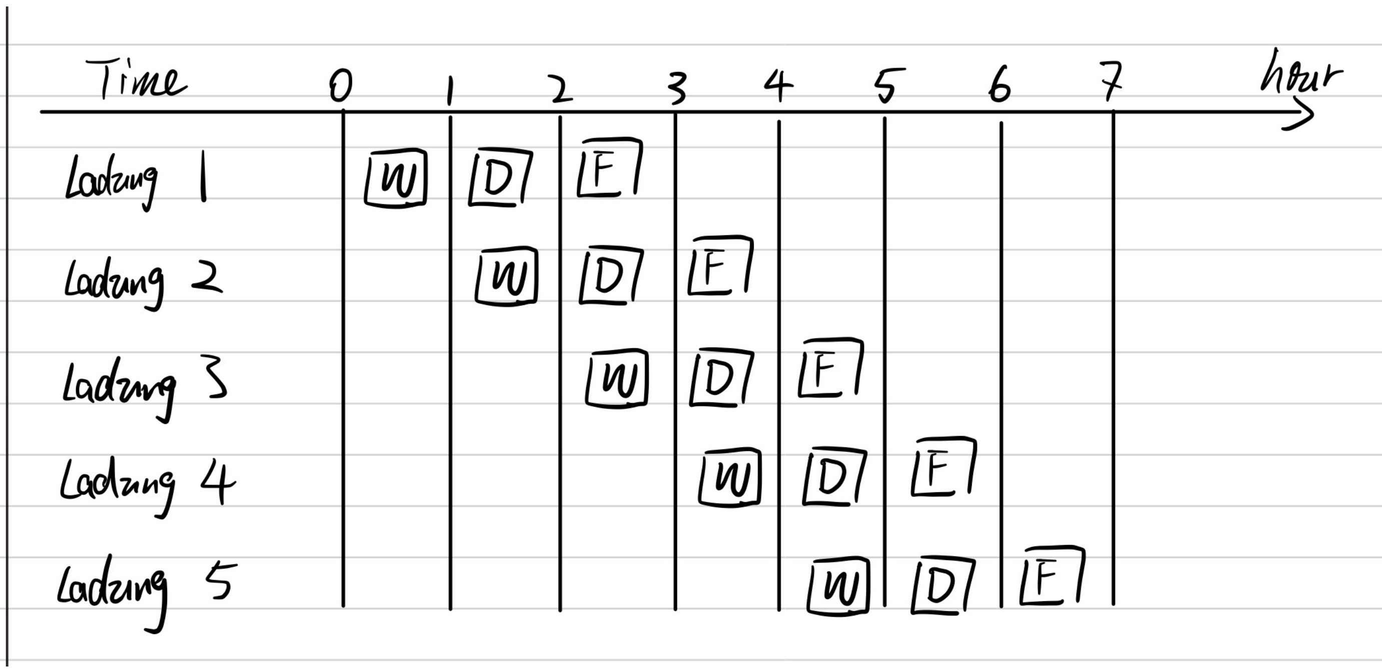
My friend and I finished a load at the same time. This means that 2 loads can be completed in 3 hours, and 6 hours to complete 4 loads.The last load takes 3 hours：

Data decomposition, because we want to make a problem’s data be decomposed into units that can be operated on relatively independently.

2 loads can be washed at the same time. T = 1 hour * (washing machine + dryer + flatiron) *⌈ (5 wash loads)/2 ⌉ = 1 * 3 * 3 = 9 hour

c)

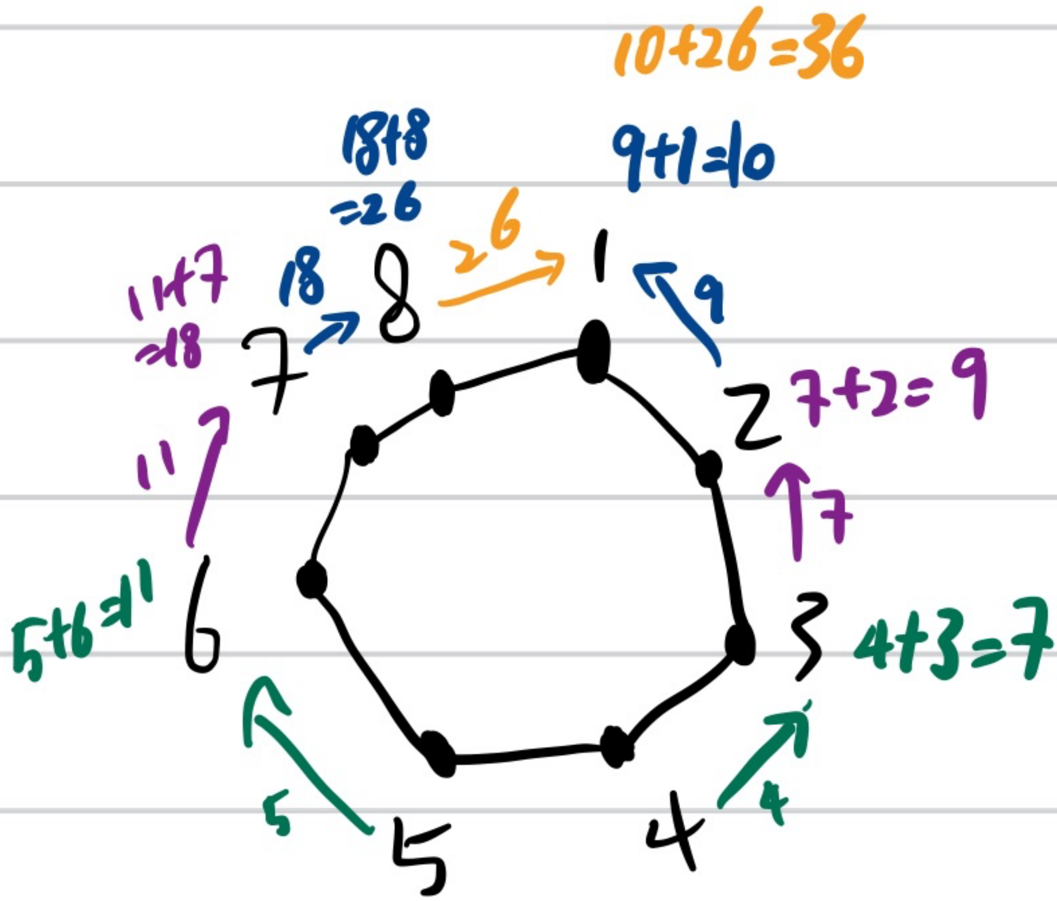
Pipeline pattern



Because the washer, dryer and the iron can be used at the same time. So after the first load uses the washer, the second load can use the washer while the first load uses the dryer. The third load can use the washer when the first load uses the iron and the second load uses the dryer. Using the parallel approach shown in the figure, all loads are completed after **7 hours**.

Aufgabe 2

a)



We assume that 8 people calculate numbers from 1 to 8:

Numbers 5 and 4 are send to numbers 6 and 3 at the same time. numbers 6 and 3 complete the addition of their numbers and the numbers received. Each gets 11 and 7.

Numbers 6 and 3 are send to numbers 7 and 2 at the same time. numbers 7 and 2 complete the addition of their numbers and the numbers received. Each gets 18 and 9.

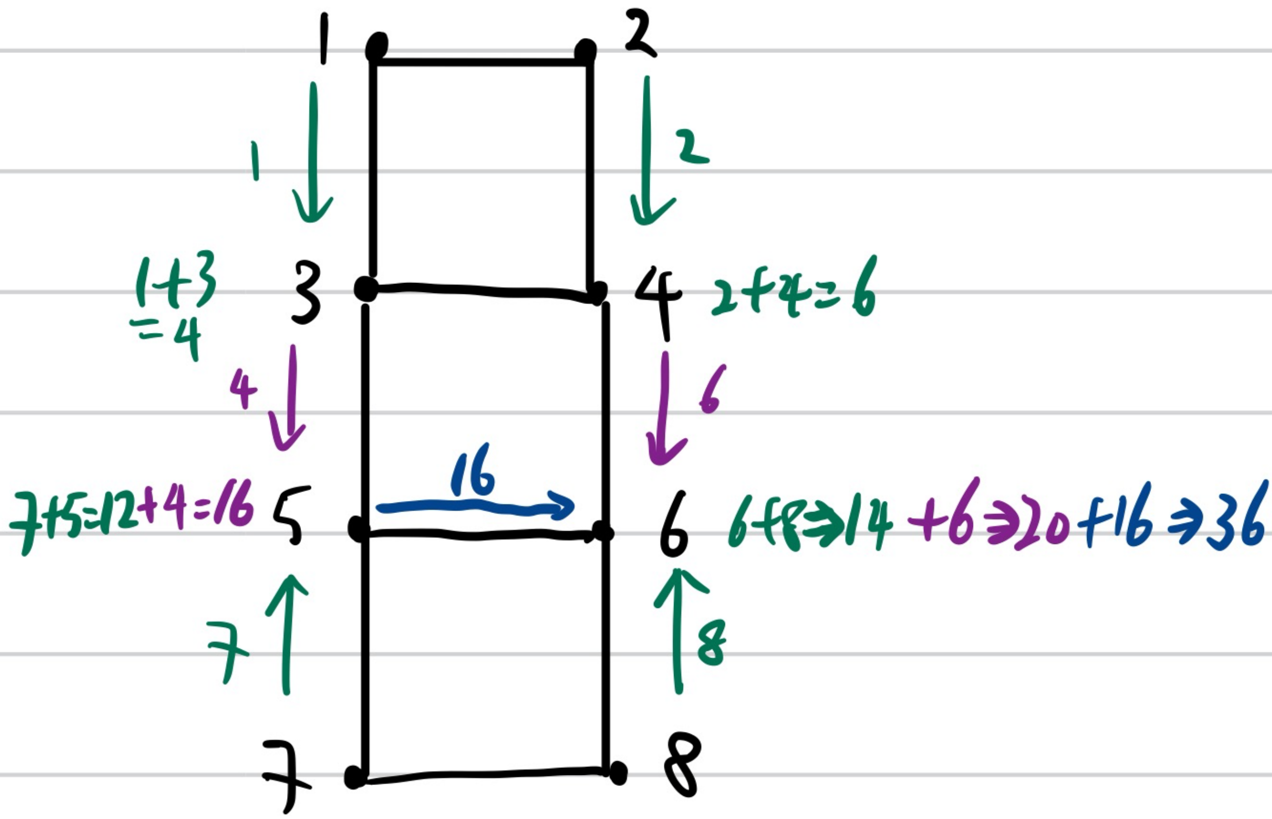
Numbers 7 and 2 are send to numbers 8 and 1 at the same time. numbers 8 and 1 complete the addition of their numbers and the numbers received. Each gets 26 and 10.

Numbers 8 send to numbers 1. numbers 1 complete the addition of its numbers and the numbers received. The final result is 10+26=36

So the total cost was:

T = $4 * t_a + 4 * t_c$

b)



Numbers 1, 2, 7 and 8 are send to numbers 3, 4, 5 and 6 at the same time. numbers 3, 4, 5 and 6 complete the addition of their numbers and the numbers received. Each gets 4 ,6, 12 and 14.

Numbers 3 and 4 are send to numbers 5 and 6 at the same time. numbers 5 and 6 complete the addition of their numbers and the numbers received. Each gets 16 and 20.

Numbers 5 send to numbers 6. numbers 6 complete the addition of its numbers and the numbers received. The final result is 20+16=36

So the total cost was:

T = $3 * t_a + 3 * t_c$

Aufgabe 3

a)

Amdahl's law:
$S = \frac{1}{(1-f_{enhanced})+\frac{f_{enhanced}}{S_{enhanced}}}$

for 1.choice:

$f_{enhanced} = 0.6$
$S_{enhanced} = 1.5$
Speedup:
$S_1 = \frac{1}{0.4+\frac{0.6}{1.5}} = \frac{1}{0.8} = 1.25$

for 2.choice:

$f_{enhanced} = 0.6/4 = 0.15$
$S_{enhanced} = 8$
Speedup:
$S_2 = \frac{1}{0.85+\frac{0.15}{8}} = 1.151$

Using Amdahl's law, we obtain the speedups S1 and S2, and since S2 is smaller than S1, acceleration of the performance of all floating point operations is better for this application.

b)

Amdahl's law:
$S = \frac{1}{(1-f_{enhanced})+\frac{f_{enhanced}}{S_{enhanced}}}$

90% execution time parallel
$f_{enhanced} = 0.9$
by 16 processors:
$S_{enhanced} = 16$
Speedup:
$S_{max} = \frac{1}{0.1+\frac{0.9}{16}} = 6.4$
Therefore, the maximum Speedup is 6.4

c)

Assume that x execution times in this application can be parallel.

$S_{enhanced} \leq 16$
Speedup:
$\frac{1}{(1-x)+\frac{x}{16}} = 10$
$\Rightarrow x = 0.96$

So, if speedup need to reach 10, at least 96% execution time in this application must be able to become parallel.