COMP3230 assignment 2

Name:Wong Ka Ngai UID:3035568881

Q1

```
knwong@workbench:~/COMP3230-Assignment-2-2020-src/q1$ ./tesla_factory.out 3 20 1 2 3
Name: Wong Ka Ngai       UID: 3035568881
Production goal: 3, num space: 20, num typeA: 1, num typeB: 2, num typeC: 3
====Final Report====
Num free space: 20
Produced Skeleton: 0
Produced Engine: 0
Produced Chassis: 0
Produced Body: 0
Produced Window: 0
Produced Tire: 0
Produced Battery: 0
Produced Car: 3
Production task completed! 3 cars produced.
Production time: 34.001972
knwong@workbench:~/COMP3230-Assignment-2-2020-src/q1$
```

how many units of storage space can be considered as sufficient for sure?

16. We need to have 7 windows, 1 body, 4 Tires, 1 battery to make a car. For making a body, 1 skeleton, 1 engine and 1 chassis is needed. So it must be sufficient to make a body with 4 space. Making a car does not require a space, so storage space=7+4+4+1=16. 16 must be sufficient for sure.

Is it always true that as the number of robot increases, the production time decreases?

No. when the number of robot increases, the extra robot may take tasks that they are not good at before other robots that perform better.
For example, imagine a case that there is no Type C robot in production. They Body part is produced by a Type A/B robot (time 4 or 3). If a Type C robot is added and it becomes responsible for the production of Body part, the time becomes 6. Keeping the production time of all other parts unchanged, the total production time increases.

How can the number of cars and the number of robots of each type affect the production time(open-ended question, share your thoughts)?

knwong@workbench.cs.hku.hk:22 - Bitvise xterm - knwong@workbench: ~/COMP3230-Assignment-2-2020-src/q1

```
Production time: 15.000525
knwong@workbench:~/COMP3230-Assignment-2-2020-src/q1$ ./tesla_factory.out 1 20 9 0 0
Name: Wong Ka Ngai       UID: 3035568881
Production goal: 1, num space: 20, num typeA: 9, num typeB: 0, num typeC: 0
====Final Report====
Num free space: 20
Produced Skeleton: 0
Produced Engine: 0
Produced Chassis: 0
Produced Body: 0
Produced Window: 0
Produced Tire: 0
Produced Battery: 0
Produced Car: 1
Production task completed! 1 car produced.
Production time: 15.000668
knwong@workbench:~/COMP3230-Assignment-2-2020-src/q1$ ./tesla_factory.out 1 20 9 0 0
Name: Wong Ka Ngai       UID: 3035568881
Production goal: 1, num space: 20, num typeA: 9, num typeB: 0, num typeC: 0
====Final Report====
Num free space: 20
Produced Skeleton: 0
Produced Engine: 0
Produced Chassis: 0
Produced Body: 0
Produced Window: 0
Produced Tire: 0
Produced Battery: 0
Produced Car: 1
Production task completed! 1 car produced.
Production time: 15.000535
knwong@workbench:~/COMP3230-Assignment-2-2020-src/q1$
```

knwong@workbench.cs.hku.hk:22 - Bitvise xterm - knwong@workbench: ~/COMP3230-Assignment-2-2020-src/q1

```
Production time: 13.000542
knwong@workbench:~/COMP3230-Assignment-2-2020-src/q1$ ./tesla_factory.out 1 20 0 9 0
Name: Wong Ka Ngai       UID: 3035568881
Production goal: 1, num space: 20, num typeA: 0, num typeB: 9, num typeC: 0
====Final Report====
Num free space: 20
Produced Skeleton: 0
Produced Engine: 0
Produced Chassis: 0
Produced Body: 0
Produced Window: 0
Produced Tire: 0
Produced Battery: 0
Produced Car: 1
Production task completed! 1 car produced.
Production time: 13.000460
knwong@workbench:~/COMP3230-Assignment-2-2020-src/q1$ ./tesla_factory.out 1 20 0 9 0
Name: Wong Ka Ngai       UID: 3035568881
Production goal: 1, num space: 20, num typeA: 0, num typeB: 9, num typeC: 0
====Final Report====
Num free space: 20
Produced Skeleton: 0
Produced Engine: 0
Produced Chassis: 0
Produced Body: 0
Produced Window: 0
Produced Tire: 0
Produced Battery: 0
Produced Car: 1
Production task completed! 1 car produced.
Production time: 13.000566
knwong@workbench:~/COMP3230-Assignment-2-2020-src/q1$
```

```
knwong@workbench.cs.hku.hk:22 - Bitvise xterm - knwong@workbench: ~/COMP3230-Assignment-2-2020-src/q1
Production time: 14.035078
knwong@workbench:~/COMP3230-Assignment-2-2020-src/q1$ ./tesla_factory.out 1 20 0 0 9
Name: Wong Ka Ngai       UID: 3035568881
Production goal: 1, num space: 20, num typeA: 0, num typeB: 0, num typeC: 9
====Final Report====
Num free space: 20
Produced Skeleton: 0
Produced Engine: 0
Produced Chassis: 0
Produced Body: 0
Produced Window: 0
Produced Tire: 0
Produced Battery: 0
Produced Car: 1
Production task completed! 1 car produced.
Production time: 14.000720
knwong@workbench:~/COMP3230-Assignment-2-2020-src/q1$ ./tesla_factory.out 1 20 0 0 9
Name: Wong Ka Ngai       UID: 3035568881
Production goal: 1, num space: 20, num typeA: 0, num typeB: 0, num typeC: 9
====Final Report====
Num free space: 20
Produced Skeleton: 0
Produced Engine: 0
Produced Chassis: 0
Produced Body: 0
Produced Window: 0
Produced Tire: 0
Produced Battery: 0
Produced Car: 1
Production task completed! 1 car produced.
Production time: 14.000532
knwong@workbench:~/COMP3230-Assignment-2-2020-src/q1$ ./tesla_factory.out 1 20 0 0 9
Name: Wong Ka Ngai       UID: 3035568881
Production goal: 1, num space: 20, num typeA: 0, num typeB: 0, num typeC: 9
```

```
knwong@workbench.cs.hku.hk:22 - Bitvise xterm - knwong@workbench: ~/COMP3230-Assignment-2-2020-src/q1
Production time: 14.000635
knwong@workbench:~/COMP3230-Assignment-2-2020-src/q1$ ./tesla_factory.out 1 20 3 3 3
Name: Wong Ka Ngai       UID: 3035568881
Production goal: 1, num space: 20, num typeA: 3, num typeB: 3, num typeC: 3
====Final Report====
Num free space: 20
Produced Skeleton: 0
Produced Engine: 0
Produced Chassis: 0
Produced Body: 0
Produced Window: 0
Produced Tire: 0
Produced Battery: 0
Produced Car: 1
Production task completed! 1 car produced.
Production time: 14.000649
knwong@workbench:~/COMP3230-Assignment-2-2020-src/q1$ ./tesla_factory.out 1 20 3 3 3
Name: Wong Ka Ngai       UID: 3035568881
Production goal: 1, num space: 20, num typeA: 3, num typeB: 3, num typeC: 3
====Final Report====
Num free space: 20
Produced Skeleton: 0
Produced Engine: 0
Produced Chassis: 0
Produced Body: 0
Produced Window: 0
Produced Tire: 0
Produced Battery: 0
Produced Car: 1
Production task completed! 1 car produced.
Production time: 14.000739
knwong@workbench:~/COMP3230-Assignment-2-2020-src/q1$
```

With 9 robots, the production time of one car varies. 9 Type A uses 15s, 9 Type B uses 13s, 9 Type C uses 14s. If we use 3 Type A, 3 Type B and 3 Type C, we get a mean result of 14s. So if the number of a specific type of robot is much more than other types, the production time could increase/decrease. So I think the safest approach is to evenly distribute the number of robots.

```
knwong@workbench.cs.hku.hk:22 - Bitvise xterm - knwong@workbench: ~/COMP3230-Assignment-2-2020-src/q1
Produced Body: 0
Produced Window: 0
Produced Tire: 0
Produced Battery: 0
Produced Car: 2
Production task completed! 2 cars produced.
Production time: 18.002296
knwong@workbench:~/COMP3230-Assignment-2-2020-src/q1$ ./tesla_factory.out 3 20 3 3 3
Name: Wong Ka Ngai       UID: 3035568881
Production goal: 3, num space: 20, num typeA: 3, num typeB: 3, num typeC: 3
====Final Report====
Num free space: 20
Produced Skeleton: 0
Produced Engine: 0
Produced Chassis: 0
Produced Body: 0
Produced Window: 0
Produced Tire: 0
Produced Battery: 0
Produced Car: 3
Production task completed! 3 cars produced.
Production time: 21.001029
knwong@workbench:~/COMP3230-Assignment-2-2020-src/q1$ ./tesla_factory.out 4 20 3 3 3
Name: Wong Ka Ngai       UID: 3035568881
Production goal: 4, num space: 20, num typeA: 3, num typeB: 3, num typeC: 3
====Final Report====
Num free space: 20
Produced Skeleton: 0
Produced Engine: 0
Produced Chassis: 0
Produced Body: 0
Produced Window: 0
Produced Tire: 0
Produced Battery: 0
Produced Car: 4
Production task completed! 4 cars produced.
Production time: 27.001172
knwong@workbench:~/COMP3230-Assignment-2-2020-src/q1$ ./tesla_factory.out 5 20 3 3 3
Name: Wong Ka Ngai       UID: 3035568881
Production goal: 5, num space: 20, num typeA: 3, num typeB: 3, num typeC: 3
====Final Report====
Num free space: 20
Produced Skeleton: 0
Produced Engine: 0
Produced Chassis: 0
Produced Body: 0
Produced Window: 0
Produced Tire: 0
Produced Battery: 0
Produced Car: 5
Production task completed! 5 cars produced.
Production time: 33.001525
knwong@workbench:~/COMP3230-Assignment-2-2020-src/q1$
```

The number of cars increases, more parts are needed so the production time increase. Using 9 robots (3 for each type), production time increase with the number of cars. From one car 14s, then two car 18s, then three cars 21s, four cars 27s, five cars 33s.

Q2

The deadlock problem arises because there is not enough storage space. A robot requires a storage space to make a body. If it gets a space, it can then consume the skeleton, engine and chassis made by other robots. However, with limited storage space, if skeleton, engine and chassis are made and occupied all the spaces before the body-making robot acquires a space, body-making robot fails to start making body so skeleton, engine and chassis cannot be consumed and they occupy the space forever. It becomes a deadlock.

To prevent this deadlock, we must ensure the body-making robot acquires a space before other robots. If the body-making robot comes first, then robots that make body subparts come after, all body subparts produced can be consumed by the body-making robot. Body is consumed by car-making robot, there is a car-making robot when the body is produced, the body will be consumed immediately and release space. Other car subparts (battery, windows, and tires) are also consumed by the car-making robot. So if car-making robot comes first, then robots that make car subparts comes after, the subparts can be consumed immediately right after production to release space.

The order is as follows,

Start to make Body -> Skeleton made-> Engine made -> Chassis made ->
Body made -> Start to make Car ->Windows, Tires, battery made -> Car made

If there are multiple cars, right after a Car is made, it starts to make body of another car, the order is looped.

To achieve this, first I changed the queue sequence to follow the order. Then I used a semaphore for each job respectively. Only the semaphore for body is initialized as 1, the other 7 are initialized as 0. Every case is started with a sem_wait so that jobs can start after a sem_post.
First, case BODY waits successfully, it runs makeBody() and post to skeleton. Skeleton waits successfully and run makeSkeleton(). After skeleton is made, it posts to engine. Engine waits successfully, engine is made and post to chassis. Chassis waits successfully, chassis is made, the body-making robot gathers all three subparts and a body is made. After a body is made, it posts to car so case CAR waits successfully. It runs makeCar() to consume the body made and posts 7 times to window, 4 times to tire and 1 time to battery. They all wait successfully and produces 7 windows, 4 tires and 1 battery. The car-making robot consumes them and finally a car is made. After a car is made, it posts to body to start all over again. The process continues to loop until all cars are made.

Using this approach, Deadlock prevention is achieved and we can ensure that it is deadlock free.

Q3

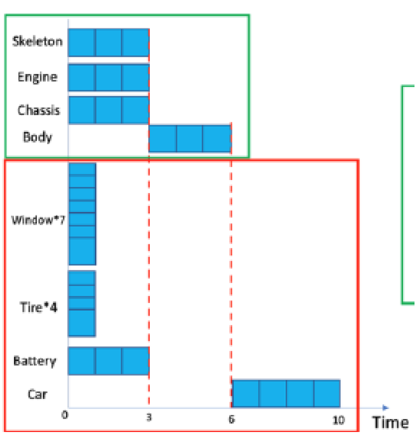
To get to the theoretical shortest production time, all parts are produced in parallel and each part is produced by the robot whose production time for that part is the shortest. In other words, we need sufficient amount of storage space and enough robots so that every job can be done by the best robot. So, if we have at least 16 storage space, 9 Type A robots, 1 Type B robot and 7 type C robot and all robots are working for the best job, that must be sufficient to achieve the theoretical shortest production time.

I optimize my algorithm using the above constraints. If we have at least 16 storage space, 9 Type A robots, 1 Type B robot and 7 type C robot, the program runs another approach to achieve the theoretical shortest production time. If the constraints does not met, approach from Q2 is used.

For the new approach, I make three queues, each for a type of robot. Only jobs that they are good at will be enqueued to their queue. For instance, Type B is good at making body only, so the queue for Type B only contains jobs of making body. When we assign a job to a robot, we check its type, dequeue its queue and assign the job dequeued to that robot. So, all robots are working for the best job.

As space is enough, order is not a big concern. So when case BODY waits successfully, it post to the three body subparts and car, then start makeBody(). Car waits successfully, post to windows, tire, battery and body, then start makeCar(). Due to the early post of body, body subparts of another car can start earlier once there is free space. The process loops until all cars are made.

For one car, the process is exactly like this, as all jobs are assigned by the best robot and they work parallelly.




```
knwong@workbench:~/COMP3230-Assignment-2-2020-src/q3$ ./tesla_
factory.out 1 16 9 1 7
Name: Wong Ka Ngai       UID: 3035568881
Production goal: 1, num space: 16, num typeA: 9, num typeB: 1,
 num typeC: 7
====Final Report====
Num free space: 16
Produced Skeleton: 0
Produced Engine: 0
Produced Chassis: 0
Produced Body: 0
Produced Window: 0
Produced Tire: 0
Produced Battery: 0
Produced Car: 1
Production task completed! 1 car produced.
Production time: 10.001043
knwong@workbench:~/COMP3230-Assignment-2-2020-src/q3$
```

scalability analysis

```
knwong@workbench:~/COMP3230-Assignment-2-2020-src/q3$ ./tesla_
factory.out 1 16 20 20 20
Name: Wong Ka Ngai       UID: 3035568881
Production goal: 1, num space: 16, num typeA: 20, num typeB: 2
0, num typeC: 20
====Final Report====
Num free space: 16
Produced Skeleton: 0
Produced Engine: 0
Produced Chassis: 0
Produced Body: 0
Produced Window: 0
Produced Tire: 0
Produced Battery: 0
Produced Car: 1
Production task completed! 1 car produced.
Production time: 10.033560
```

keeping the space same as previous test (16), if we increase the number of robots (each 20), the time is still the same (10s). That's because at most only 17 robots is needed to produce a car.

```
knwong@workbench:~/COMP3230-Assignment-2-2020-src/q3$ ./tesla_
factory.out 1 60 20 20 20
Name: Wong Ka Ngai       UID: 3035568881
Production goal: 1, num space: 60, num typeA: 20, num typeB: 2
0, num typeC: 20
====Final Report====
Num free space: 60
Produced Skeleton: 0
Produced Engine: 0
Produced Chassis: 0
Produced Body: 0
Produced Window: 0
Produced Tire: 0
Produced Battery: 0
Produced Car: 1
Production task completed! 1 car produced.
Production time: 10.002233
```

If we increase the space (60), the time is still the same (10s). That's because at most only 17 robots is needed to produce a car.

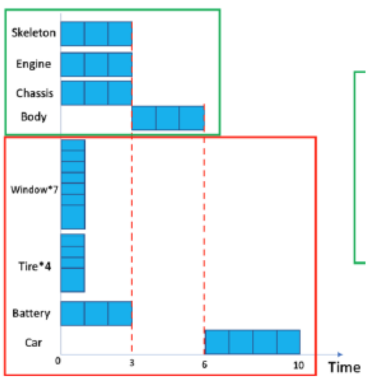
```
knwong@workbench:~/COMP3230-Assignment-2-2020-src/q3$ ./tesla_
factory.out 2 16 9 1 7
Name: Wong Ka Ngai       UID: 3035568881
Production goal: 2, num space: 16, num typeA: 9, num typeB: 1,
 num typeC: 7
====Final Report====
Num free space: 16
Produced Skeleton: 0
Produced Engine: 0
Produced Chassis: 0
Produced Body: 0
Produced Window: 0
Produced Tire: 0
Produced Battery: 0
Produced Car: 2
Production task completed! 2 cars produced.
Production time: 16.051854
```

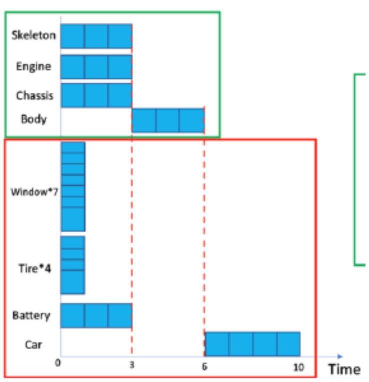
When we increase the number of cars to 2, 16 space with 17 robots need 16s to make two cars. The time is not 10*2=20s because robots did not wait until the first car is finished to produce parts of the second car. When they finish a job, they pick up jobs from the second car immediately. So it is faster than 20s.

```
knwong@workbench:~/COMP3230-Assignment-2-2020-src/q3$ ./tesla_
factory.out 2 16 20 20 20
Name: Wong Ka Ngai       UID: 3035568881
Production goal: 2, num space: 16, num typeA: 20, num typeB: 2
0, num typeC: 20
====Final Report====
Num free space: 16
Produced Skeleton: 0
Produced Engine: 0
Produced Chassis: 0
Produced Body: 0
Produced Window: 0
Produced Tire: 0
Produced Battery: 0
Produced Car: 2
Production task completed! 2 cars produced.
Production time: 12.043275
```

keeping the space same as previous test (16), if we increase the number of robots (each 20), it becomes faster (12s). That's because extra robots acquire jobs quickly and they can start working once a space is released. Without the extra robots, robots need to finish their job, then acquire a new job, then wait for a space to perform the new job. Therefore, even storage space is the same, more robots makes it faster.

```
knwong@workbench:~/COMP3230-Assignment-2-2020-src/q3$ ./tesla_
factory.out 2 60 20 20 20
Name: Wong Ka Ngai        UID: 3035568881
Production goal: 2, num space: 60, num typeA: 20, num typeB: 2
0, num typeC: 20
====Final Report====
Num free space: 60
Produced Skeleton: 0
Produced Engine: 0
Produced Chassis: 0
Produced Body: 0
Produced Window: 0
Produced Tire: 0
Produced Battery: 0
Produced Car: 2
Production task completed! 2 cars produced.
Production time: 10.002805
```





With more space (60) and more robots (each 20), it needs only 10s to make two cars. That’s because the production of two cars are parallel. Just like the figure on the left.

```
knwong@workbench:~/COMP3230-Assignment-2-2020-src/q3$ ./tesla_
factory.out 7 112 63 7 49
Name: Wong Ka Ngai       UID: 3035568881
Production goal: 7, num space: 112, num typeA: 63, num typeB:
7, num typeC: 49
====Final Report====
Num free space: 112
Produced Skeleton: 0
Produced Engine: 0
Produced Chassis: 0
Produced Body: 0
Produced Window: 0
Produced Tire: 0
Produced Battery: 0
Produced Car: 7
Production task completed! 7 cars produced.
Production time: 10.023801
```

```
knwong@workbench:~/COMP3230-Assignment-2-2020-src/q3$ ./tesla_
factory.out 10 160 90 10 70
Name: Wong Ka Ngai       UID: 3035568881
Production goal: 10, num space: 160, num typeA: 90, num typeB:
 10, num typeC: 70
====Final Report====
Num free space: 160
Produced Skeleton: 0
Produced Engine: 0
Produced Chassis: 0
Produced Body: 0
Produced Window: 0
Produced Tire: 0
Produced Battery: 0
Produced Car: 10
Production task completed! 10 cars produced.
Production time: 10.006290
```

For making n cars, if we have at least 16*n storage space, 9*n Type A robot, n Type B robot and 7*n Type C robot, we can predict that the time will be around 10s.

Making 7 cars or 10 cars need 10s only, as long as the above requirement is fulfilled.

```
knwong@workbench:~/COMP3230-Assignment-2-2020-src/q3$ ./tesla_
factory.out 1 2 3 0 1
Name: Wong Ka Ngai        UID: 3035568881
Production goal: 1, num space: 2, num typeA: 3, num typeB: 0,
num typeC: 1
====Final Report====
Num free space: 2
Produced Skeleton: 0
Produced Engine: 0
Produced Chassis: 0
Produced Body: 0
Produced Window: 0
Produced Tire: 0
Produced Battery: 0
Produced Car: 1
Production task completed! 1 car produced.
Production time: 35.007828
```

For making a car, if we do not have 16 storage space, 9 Type A robots, 1 Type B robot and 7 type C robot, approach in Q2 will be used to avoid deadlock happening from not having enough space. Just like the screenshot above, having 2 space and 4 robots uses Q2 approach to deal with deadlock.

This program can be further optimized. Actually I guess we can use less than 16 storage space, 9 Type A robots, 1 Type B robot and 7 type C robot to get to the theoretical shortest production time. For example: