

TOYATA PRIUS

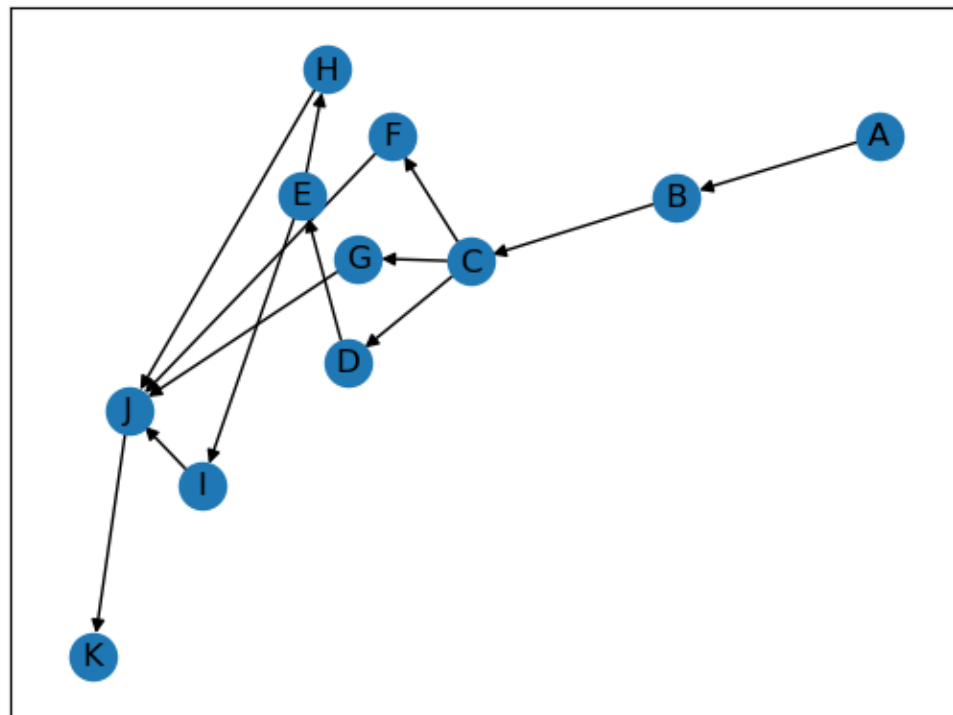


Fərz edək ki, 800 toyota prius zavodda yığılmalıdır. Gündə beş yüz prius tələb olunur. Gündə istehsal müddəti 420 dəqiqədir və montaj mərhələləri və priusun istehsal müddəti aşağıda verilmişdir. Dövr vaxtı və prioritet məhdudiyyətləri nəzərə alınmaqla iş stansiyalarının sayını minimuma endirən bir tarazlıq əldə edək.



Tapşırıq	Tapşırıq vaxtı (saniyələrlə)	Təsvir	Birinci yerdə olmalı olan tapşırıqlar
A	45	Arxa ox dəstəyi yerləşdirin və əllə bərkidin	-
B	11	4 bolt qaykanın bərkildilməsi	A
C	9	Arxa oxu daxil etmək	B
D	50	Arxa oxu dəstəkləyən vintləri qaykalara bərkidin	-
E	15	Ön ox montajını və əlini yerləşdirmək	D
F	12	Dörd vint ilə qayka-boltları bərkitmək	C
G	12	Ön ox montaj vintlərini sıxın	C
H	12	Arxa təkəri 1 yerləşdirin və qapağı bərkidin	E
J	12	Arxa təkər 2-ni yerləşdirin və qapağı bərkidin	E
J	8	Ön təkəri 1 yerləşdirin və qapağı bərkidin	F,G,H,I
K	9	Ön təkəri 2 yerləşdirin və qapağı bərkidin	J

1. Prioritet diaqramını aşağıdakı kimi çəkək:



2. İş stansiyasının dövriyyə müddətini təyin edək. Burada istehsal vaxtını saniyələrə çevirməliyik, çünki tapşırıq vaxtlarımız saniyələrlədir.

$$C = \frac{\text{Gündə istehsal vaxtı}}{\text{Gündə tələb olunan çıxış (vahidlərlə)}} = \frac{420 \text{ dəq} * 60 \text{ saniyə}}{500 \text{ avtomobil}} = \frac{25200}{500} = 50.4 \text{ saniyə}$$

MÜƏLLİF: İBRAHİM ABBASOV

```
# Import necessary libraries
import networkx as nx
import matplotlib.pyplot as plt

# İstiqamətləndirilmiş Qrafik yaradaq
G = nx.DiGraph()

# Tapşırıqları və onlara uyğun tapşırıq vaxtlarını müəyyənləşdirək.
tasks = {
    'A': 45,
    'B': 11,
    'C': 9,
    'D': 50,
    'E': 15,
    'F': 12,
    'G': 12,
    'H': 12,
    'I': 12,
    'J': 8,
    'K': 9
}

# Tapşırıq asılılıqlarını müəyyənləşdirək.
dependencies = {
    'A': [],
    'B': ['A'],
    'C': ['B'],
    'D': ['C'],
    'E': ['D'],
    'F': ['C'],
    'G': ['C'],
    'H': ['E'],
    'I': ['E'],
    'J': ['F', 'G', 'H', 'I'],
    'K': ['J']
}

# Qrafikə tapşırıqlar və asılılıqlar əlavə edək.
for task, time in tasks.items():
    G.add_node(task, time=time)

for task, dep_tasks in dependencies.items():
    for dep_task in dep_tasks:
        G.add_edge(dep_task, task)

# Dövr müddətini hesablayaq
cycle_time = 420 # Gündə istehsal vaxtı dəqiqələrlə verilmişdir
cycle_time /= 500 # Gündə tələb olunan avtomobillərin sayına bölünərək dövrə müddətini bir avtomobilə dəqiqələrlə əldə etmək olar

# İş stansiyalarının sayını minimuma endirən balansı hesablayaq.
# Bu, Siyahı Planlaşdırma alqoritmi kimi resurs məhdud layihə planlaşdırma alqoritmlərindən istifadə etməklə həll edilə bilər.
# Bununla belə, belə bir alqoritmin həyata keçirilməsi sadə kod parçasının ahata dairəsindən kənara çıxır.

# Montaj prosesinin qrafikini görüntüləyək.
nx.draw_networkx(G, with_labels=True)
plt.show()
```

MÜƏLLİF: İBRAHİM ABBASOV

