

שאלה 4

הסבר האלגוריתם:

תחילה אצור גרף, המייצג את כלל הרכבות מהערים הנתונים. כל עיר תייצג קודקוד, כל רכבת תייצג קשת. לכן לכל $(u, v) \in E$ לא נשמור משקל, אלא נשמור זמן התחלה, כלומר זמן יציאת הרכבת, וזמן סיום. כעת נפעיל בערך את אלגוריתם דיקסטרה. אבל צריך להתחשב בדבר נוסף, האם זמן הגעת לעיר, מאפשר לי לקיחה של רכבת מסויימת. לכן כאשר נבחר קשת (רכבת) הנכנסת לעיר, נעדכן את כל הקודקודים השכנים (הערים שיש מהעיר שבחרנו רכבת אליהם), שיש דרך הגעה חדשה אליהם, רק אם נוכל לקחת את הרכבת מבחינת זמן יציאתה. נשמור לכל קודקוד במקום את השדה $dis[v]$, נשמור את $time[v]$ בכל קודקוד (כלומר כל עיר), נשמור ב $time[v]$ את זמן ההגעה לקודקוד. ולכן בסוף האלגוריתם נחזיר את $time[t]$.

פסודו קוד:

```
Q is a priority queue on time[v]
for v ∈ V:
    time[v] = ∞
    π[v] = s (s = given start city)
time[s] = X (X = given time of start)
π[v] = null

while Q.NotEmpty:
    v = Q.extractMin
    for u ∈ Adj(v):
        if time[u] ≤ start((v,u)):
            if dis[u] > end((v,u)) - time[v]:
                dis[u] = end((v,u)) - time[v]
                π[u] = v
return time[t] (t = given destination city)
```

הוכחת נכונות:

ניתוח סיבוכיות מקום וזמן ריצה:

מקום

אנו שומרים את אותם הנתונים כמו בדיקסטרה. ולכן סיבוכיות המקום של האלגוריתם הינה $O(|V|)$.

זמן

גם מבחינת זמן ריצה, אנו לא משנים כלום מדיקטורה.
ולכן זמן ריצת האלגוריתם הינה $O(|E| + |V|\log|V|)$.