

۱- (نظری) شما مسئول زمان‌بندی عملیات درمانی در اتاق اورژانس یک بیمارستان شلوغ هستید. پنج بیمار همزمان با آمبولانس به بیمارستان می‌رسند که نیاز به خدمات فوری دارند، اطلاعات آن‌ها به صورت زیر است:

#	توضیحات	زمان درمان	مهلت مناسب درمان
1	شکستگی دست	a	a + 4
2	واکنش آلرژیک	b	b + 3
3	حمله قلبی	c	c + 2
4	جراحی جزئی	d	d + 6
5	سوختگی	e	e + 5

با استفاده از ارقام شماره دانشجویی خود، جدول بالا را تکمیل نمایید. به این صورت که پنج رقم آخر شماره دانشجویی شما مشخص کننده $\overline{abcde} = 95234$ است. به جای رقم ۰، ۵ قرار دهید. به عنوان مثال شماره دانشجویی ۴۰۰۱۹۰۲۳۴ به صورت $\overline{abcde} = 95234$ خواهد بود. با استفاده از الگوریتم Earliest Due Date (EDD)، وظایف را زمان‌بندی کنید. زمان‌بندی وظایف را نشان داده و برای هر بیمار، Response time، Lateness و Laxity را بدست آورید. همچنین مشخص کنید که کدام بیمارها نمی‌توانند در زمان مناسب درمان شوند.

تمرین شماره ۲

درس سامانه‌های بی‌درنگ

۲ - نظری) شما مسئول مدیریت سیستم اعزام اضطراری در یک شهر هستید، جایی که چندین وظیفه اضطراری (مانند تماس‌های آمبولانس، عملیات نجات آتش‌نشانی و مداخلات پلیس)، باید زمان‌بندی شوند. این وظایف به صورت پویا وارد می‌شوند و به دلیل ماهیت بحرانی عملیات، شما باید یک زمان‌بندی non-preemptive برای این وظایف انجام دهید که موارد اضطراری به موقع رسیدگی شوند. جزئیات وظایف به صورت زیر است:

#	توضیحات	زمان تماس	زمان اجرای عملیات	مهلت زمانی
1	اعزام آمبولانس	a	4	7 + a
2	عملیات نجات آتش‌نشانی	2	4	10
3	مداخله پلیس	3	3	15
4	وضعیت اورژانسی پزشکی	4	2	12

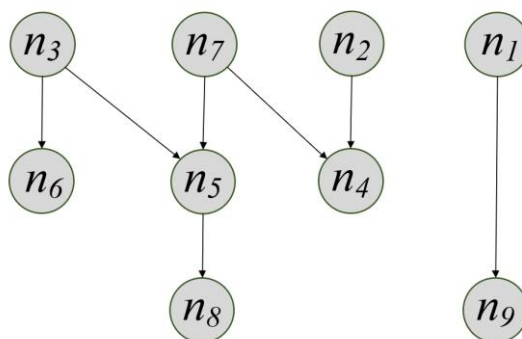
با استفاده از رقم آخر شماره دانشجویی خود، جدول بالا را تکمیل نمایید. به عنوان مثال شماره دانشجویی ۴۰۰۱۹۰۲۳۴ به صورت $a = 4$ خواهد بود. با استفاده از الگوریتم Bratley یک نمایش درختی از زمان‌بندی‌های ممکن ترسیم کنید. همچنین مشخص کنید آیا زمان‌بندی وجود دارد که مهلت زمانی تمام وظایف رعایت شود یا خیر.

۳ - نظری) مجموعه وظایف زیر را در نظر بگیرید. اولویت اجرای وظایف به صورت نزولی (بر حسب اندیس آن‌ها است)، به عبارتی اگر $i < j$ ، آنگاه اولویت n_i بیشتر از n_j است. ابتدا وظایف را بر روی ۳ پردازنده زمان‌بندی کنید به طوری که وظیفه با بیشترین اولویت روی اولین پردازنده در دسترس اجرا شود (در صورت در دسترس بودن چند پردازنده، پردازنده با اندیس کمتر اولویت دارد). سپس بررسی کنید در کدام یک از حالت‌های زیر آنومالی رخ می‌دهد.

- زمان اجرای تمام وظایف یک واحد کم شود.
- یک پردازنده به پردازنده‌ها اضافه شود.
- محدودیت بین وظایف ۲-۴ و ۴-۷ حذف شود.

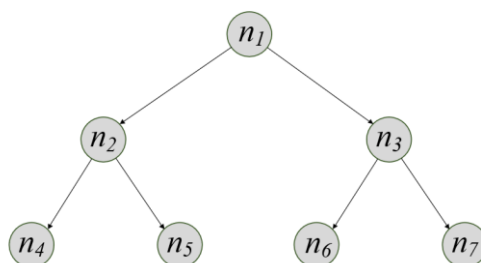
دلیل بررسی خود را نیز بیان کنید.

وظیفه	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8	n_9
زمان اجرا	2	8	8	30	30	30	6	9	10



شکل ۱، گراف وظایف سامانه

۴ - پیاده‌سازی) در یک سامانه بی‌درنگ تعدادی وظیفه به هم وابسته داریم، این وظایف به صورت گراف وظایف مدل می‌شوند. یک نمونه از گراف وظایف در شکل ۲ آورده شده است.



شکل ۲، نمونه گراف وظایف

در این سؤال شما باید یک زمان‌بند برای زمان‌بندی وظایف در چنین سامانه‌ای پیاده‌سازی کنید. برای این کار موارد زیر را در نظر بگیرید:

- از الگوریتم EDF* برای زمان‌بندی وظایف استفاده کنید.
- زمان‌بند شما به هر زبان برنامه‌نویسی‌ای می‌تواند پیاده‌سازی شود.
- ورودی برنامه شما یک فایل JSON به صورت زیر است:

```
[
  {
    "taskId": 1,
    "parentId": null,
    "releasedTime": 1,
    "executionTime": 5,
    "deadline": 10
  },
  {
    "taskId": 2,
    "parentId": 1,
    "releasedTime": 1,
    "executionTime": 4,
    "deadline": 10
  },
  {
    "taskId": 3,
    "parentId": 1,
    "releasedTime": 1,
    "executionTime": 6,
    "deadline": 20
  },
  {
    "taskId": 4,
    "parentId": 2,
    "releasedTime": 4,
    "executionTime": 2,
    "deadline": 25
  },
  {
    "taskId": 5,
    "parentId": 2,
    "releasedTime": 4,
    "executionTime": 3,
    "deadline": 25
  }
]
```

```

        "taskId": 6,
        "parentId": 3,
        "releasedTime": 5,
        "executionTime": 1,
        "deadline": 16
    },
    {
        "taskId": 7,
        "parentId": 3,
        "releasedTime": 32,
        "executionTime": 2,
        "deadline": 40
    }
]

```

- در خروجی شما باید زمان‌های اجرا، زمان ورود تغییر یافته و مهلت زمانی تغییر یافته به ازای هر وظیفه ارائه دهید. خروجی شما باید به صورت Json و مانند زیر باشد.

```

[
    {
        "taskId": 1,
        "releasedTime": 1,
        "deadline": 9,
        "executionTimes": [[1, 6]]
    },
    {
        "taskId": 2,
        "releasedTime": 6,
        "deadline": 22,
        "executionTimes": [[13, 17]]
    },
    {
        "taskId": 3,
        "releasedTime": 6,
        "deadline": 15,
        "executionTimes": [[6, 12]]
    },
    {
        "taskId": 4,
        "releasedTime": 10,
        "deadline": 25,
        "executionTimes": [[17, 19]]
    },
    {
        "taskId": 5,
        "releasedTime": 10,
        "deadline": 25,
        "executionTimes": [[19, 22]]
    },
    {

```

```
{
  "taskId": 6,
  "releasedTime": 12,
  "deadline": 16,
  "executionTims": [[12, 13]]
},
{
  "taskId": 7,
  "releasedTime": 32,
  "deadline": 40,
  "executionTimes": [[32, 34]]
}
]
```