

پروژه TSP (۱۰۰ امتیاز)

۱- معرفی

هدف از این پروژه پیاده سازی صحیح دو رویکرد متفاوت برای حل مساله TSP و همچنین اندازه گیری زمان اجرای آن ها بر روی چندین ورودی مختلف است. فرض کنید فایل ورودی به صورت زیر باشد:

5
2 7
4 6
18 3
12 50
7 29

در خط اول n ، یعنی تعداد نقاط آمده است. سپس در هر یک از n خط بعدی، مختصات (x,y) یکی از نقاط آمده است. خروجی برنامه شما باید هم تور TSP (ترتیب پیمایش نقاط) و هم طول این تور را گزارش کند.

۱- رویکرد اول، رویکرد «نزدیک ترین همسایه» یا Nearest Neighbor است که در کلاس مطرح شد و شبه کد آن نیز در صفحه ۶ کتاب موجود است. این رویکرد سریع است اما لزوما پاسخ بهینه را پیدا نمی کند.

۲- رویکرد دوم، رویکرد «جستجوی کامل» یا Exhaustive Search است که در کلاس نیز مورد بحث قرار گرفت. این رویکرد تمامی جایگشت های ممکن را امتحان می کند و شبه کد آن نیز در صفحه ۸ کتاب موجود است. این رویکرد همواره پاسخ بهینه را پیدا می کند ولی بسیار کند است. شما می توانید از طریق اینترنت یا هر منبع دیگری کد یا الگوریتمی را پیدا کنید که تمامی جایگشت های ممکن را تولید کند.

به خاطر داشته باشید که مطابق تعریف TSP، تور می تواند از هر یک از n نقطه آغاز شود. ولی به دلیل یک پارچه بودن پاسخ ها، الگوریتم شما باید از اولین نقطه ای که در ورودی آمده است شروع کند (به عنوان مثال، در نمونه ای که آورده شده، الگوریتم شما باید از نقطه (2,7) شروع کند).

۲- موارد تحویلی

۱-۲- گزارش

۱- (۱۵ امتیاز) جزئیات هر دو پیاده سازی خود را توضیح دهید. به خصوص توضیح دهید که چگونه عبارات سطح بالا و انگلیسی در شبه کد را به صورت بهینه پیاده سازی کرده اید.

۲- (۱۵ امتیاز) پیچیدگی زمانی الگوریتم های خود در بدترین حالت را بر حسب n تعیین کنید.

۳- (۲۰ امتیاز) از یک تولید کننده اعداد تصادفی برای تولید ورودی الگوریتم های خود به ازای حداقل چهار n متفاوت استفاده کنید. مقادیر n برای هر یک از دو رویکرد ممکن است نیاز باشد که متفاوت باشد. برای انتخاب n ، موارد زیر را در نظر داشته باشید:

الف) n باید به اندازه کافی بزرگ انتخاب شود تا بتوانید زمان اجرای برنامه را به وسیله توابعی همچون $\text{clock}()$ تعیین کنید و مقداری که به دست می آید قابل اعتماد باشد. به عنوان مثال زمان اجرای برنامه باید حداقل ۱۰ برابر بیشتر از دقت تابعی که برای اندازه گیری زمان استفاده می کنید باشد (منظور از دقت تابع، کوچک ترین واحد زمانی قابل اندازه گیری توسط این تابع است).

ب) همچنین n را به گونه ای انتخاب کنید که بتوانید به صورت عملی، پیچیدگی زمانی تئوری که پیش تر به دست آورده بودید را ارزیابی کنید.

به ازای هر n ، سه مرتبه برنامه را با ورودی یکسان اجرا کنید و میانگین زمان اجرا در این ۳ مرتبه را به عنوان زمان اجرا در نظر بگیرید. این کار باعث می شود که احتمال خطای ناشی از بار روی سیستم کاهش یابد. نتایج خود را در یک جدول نمایش دهید و توضیح دهید که n را چگونه انتخاب کردید.

۴- (۱۰ امتیاز) تئوری را با نتایج عملی تطبیق دهید: نشان دهید که زمان اجرایی که در عمل به دست آوردید، با پیچیدگی زمانی که به صورت تئوری به دست آورده بودید، همخوانی دارد.

۲-۲- ارزیابی

۱- (۴۰ امتیاز) ارزیابی: پس از این که گزارش های خود را submit کردید، ما از شما می خواهیم که هر دو الگوریتم خود را با ورودی هایی که توسط ما آماده شده است اجرا کنید. منتظر اطلاع رسانی در این مورد باشید.