به نام خدا

تمرین سری چهارم
طراحی الگوریتم‌های پیشرفته

نعیم اصفهانی
۸۴۲۰۱۰۰۳

1.5

می‌توانیم این مساله را با پوشش راسی حل کنیم به این ترتیب که راس‌ها را مجموعه‌ها گرفته واعضا را یال‌ها می‌گیریم. هزینه‌ی همه‌ی راس‌ها را هم برابر می‌گیریم. عملا کار ما باعث می‌شود راسی که درجه‌ی بیشتری دارد انتخاب شود چون میانگین هزینه‌ی آن کمتر می‌شود. پس ضریب تقریب این مساله هم همان $H_n \approx \lg n$ می‌باشد. برای مثال بسته هم چون مساله به مساله‌ی پوشش راسی کاهش داده شد می‌توان از همان مثال بسته‌ی مساله‌ی پوشش راسی گراف مورد نظر را ساخت.

2.14

a.

ضریب تقریب این قسمت دقیقا برابر ضریب تقریب کوچکترین ابر رشته است. چون تنها ورودی مساله را دو برابر کرده‌ایم پس ضریب تقریب همان ضریب $2H_n$ می‌باشد.

b.

لختی(Relaxation) مطرح شده نیز بهبودی در ضریب تقریب الگوریتم ایجاد نمی‌کند چون می‌توان ورودی الگوریتم را مجموعه‌ای از رشته‌های قرینه در نظر گرفت و در این صورت خود الگوریتم با الگوریتم لخت مطرح شده یکسان است و همان ضریب تقریب را به ما می‌دهد.

3.14

همان راه حل تقریبی مربوط به TSP را پیش می‌گیریم با این تفاوت که در موقع میان‌بر زدن ابتدایی در دور اولری ابتدا یال (u,v) را استخراج می‌کنیم و سپس میان برهای دیگر را بدست می‌آوریم. در نهایت یال (u,v) در این دور را حذف کرده و مسیر ساده بدست می‌آید. بدیهی است که این راه حل جواب تقریبی برابر با الگوریتم استفاده شده برای تولید میان‌برها می‌دهد. چون تنها تغییری که ما در این روند انجام دادیم در ترتیب عمل بود. اگر از الگوریتم اول استفاده کنیم تقریب 2 و اگر از الگوریتم دوم استفاده کنیم تقریب $\frac{3}{2}$ بدست می‌آوریم.

3.15

الگوریتم خواسته شده عضو P است. راه حل این است که تمام فرستنده‌ها را به عنوان یک نود بگیریم و کوچکترین درخت فراگیر بر روی آن‌ها را تشکیل دهیم. مساله در صورتی NP است که به جای بخش بندی گراف به دو مجموعه‌ی فرستنده و گیرنده، این دو تنها زیر مجموعه‌هایی از راس‌ها باشند. به عبارتی راس‌های دیگری هم وجود داشته باشند. در این صورت با گرفتن تمام فرستنده‌ها به عنوان یک نود مساله‌ی درخت متریک اشتاینر حاصل می‌شود(این سوپر نود و گیرنده‌ها راس‌های اجباری و مابقی اختیاری می‌شوند) که می‌دانیم ضریب تقریب آن ۲ است.

4.7

اگر یال‌هایی را که در هر مرحله s_i و s_j را از هم جدا می‌کنند را A_{ij} بنامیم، بدیهی است که با حذف مجموعه‌ی $A=\bigcup_{i,j=1}^{k} A_{ij}$ همه‌ی این قسمت‌ها از هم جدا می‌شوند. با توجه به اینکه در هر مرحله عمل انجام شده روی بقیه‌ی گراف انجام می‌شود و هر یال در دو مجموعه حساب می‌شود داریم: $\sum_{i,j=1}^{k} w(A_{ij}) = 2w(A)$.

با توجه به این‌که با انتخاب کوچکترین در هر مرحله ما $k-1$ کوچکترین برش‌ها را انتخاب کردیم داریم:

$$w(C) \le \left(1-\frac{1}{k}\right)\sum_{i,j=1}^{k} w(C_{ij}) \le \left(1-\frac{1}{k}\right)\sum_{i,j=1}^{k} w(A_{ij}) \le 2\left(1-\frac{1}{k}\right)w(A)$$

$$w(C) \le \left(2-\frac{2}{k}\right)w(A)$$

5.7

ما عملا روی مساله‌ی بیشینه‌سازی معادل کار می‌کردیم و سعی می‌کردیم آن را ماکزیمال کنیم و در نتیجه هرچه بدست می‌آوردیم قطعا از بهینه کمتر بود و رابطه‌ی $j \le i^*$ را داشتیم. ولی اگر از مینیمال کردن مساله‌ی اصلی استفاده کنیم خواهیم داشت $j \ge i^*$ که مشکل ما را حل نمی‌کند و یک گزاره‌ی بدیهی است.