الگوریتم موازی برای درخت فراگیر کمینه بر روی MOT

- ورودی: یک گراف وزن‌دار (ماتریس مجاورت $n \times n$: n رأس و حداکثر $n^۲$ یال)
- خروجی: درخت فراگیر کمینه (یال‌های گراف که درخت را می‌سازند مشخص می‌شوند)
- ساختار: 2D-MOT که رأس‌های درخت سطر i و ستون i در هم ادغام شده باشند.
- پردازنده‌ی (i, j) در توری متناظر است با یال (i, j) که در ابتدا وزن آن یال $w(i, j)$ را ذخیره می‌کند.
- ریشه‌ی i متناظر است با رأس i

- رأس‌ها افراز می‌شوند به مجموعه‌هایی که یکی از اعضای آن مجموعه leader آن است. هر رأس i leader خود را در $L(i)$ ذخیره می‌کند. اگر $i = L(i)$ باشد i leader است. مجموعه‌ای که i leader آن است را S_i می‌گوییم.
-
- در ابتدا هر رأس به تنهایی یک مجموعه است. در انتها فقط یک مجموعه داریم که یال‌های آن MST است.

قضیه: سبک‌ترین یال که دو رأس از دو مجموعه‌ی مختلف را به‌هم وصل می‌کند جزء MST است.

کلیات الگوریتم

- $\lg n$ بار تکرار کن (تا این‌که فقط یک مجموعه داشته باشیم)
- به صورت موازی هر leader نزدیک‌ترین leader به خود را پیدا می‌کند.
- هر leader مجموعه‌اش را در نزدیک‌ترین مجموعه ادغام می‌کند.
- انجام کار فوق به صورت موازی موجب ادغام تعدادی مجموعه و تولید مجموعه‌های بزرگ‌تر می‌کند. تعداد مجموعه‌ها حداقل نصف می‌شود.
- پس از ادغام باید leader جدید انتخاب شود.

پیدا کردن $P(i)$ نزدیک‌ترین leader به leader شماره‌ی i

فرض: $W(i)$ وزن سبک‌ترین یال بین S_i و $S_{P(i)}$

- رأس i، $L(i)$ را برای برگ‌ها درخت افقی و عمودی می‌فرستد.
- پردازنده‌ی (i, j)، مقادیر $L(i)$، $L(j)$، $W(i, j)$ را دارد.
- بر روی درخت ستونی، اگر $L(j) \neq L(i)$ باشد $W(i, j)$، $L(i)$، i و j به بالا ارسال می‌شود و مینیمم گرفته می‌شود.
- رأس j سبک‌ترین یالی که یک‌سرش S_j است و یک‌سر دیگرش S_k که $k \neq j$ را به دست می‌آورد.
- اگر $i = L(i)$، i مقادیر $P(i)$ و $W(i)$ را ذخیره می‌کند.
- این کار در $2 \lg n$ مرحله انجام می‌شود.

ادغام مجموعه‌ها

- اگر $(i, P(i))$ را یال‌های یک گراف جهت‌دار $\mathcal{G}$ در نظر بگیریم،
- هر مسیری را که در $\mathcal{G}$ دنبال کنیم به یک حلقه بین (i, j) می‌رسیم که $i = P(j)$ و $j = P(i)$
- leader جدید رأس با کم‌ترین شماره در هر حلقه انتخاب می‌شود.
- اگر $\lg n$ بار عمل $P(i) \leftarrow P(P(i))$ را انجام دهیم، هر leader i به leader جدید می‌رسد. یعنی $P(i)$ leader جدید همه‌ی رأس‌هایی است که i قبلاً leader آن بوده است.
- برای این کار باید در اولین گام در هر حلقه‌ی (i, j) اگر i leader نهایی است، $P(i) = i$ و $P(j) = i$ ذخیره شود.

- در انتـهـا کـافـی اسـت کـه هـر رأس i (هـم leader و هـم بـقـیـه‌ی رأس‌هـا) عـمـل $L(i) \leftarrow P(L(i))$ را انجام دهد.
- نشان می‌دهیم که عمل‌های $P(i) \leftarrow P(P(i))$ و $L(i) \leftarrow P(L(i))$ را می‌توان در $۲ \lg n$ مرحله انجام داد.

پس در مجموع الگوریتم $O(\lg^۳ n)$ است. این الگوریتم را می‌توان در $O(\lg^۲ n)$ هم انجام داد. به کتاب لایتون مراجعه کنید.

انجام عمل $Z(i) = X(Y(i))$

- $X(i)$ به برگ‌های درخت سطر i ارسال می‌شود.
- در همان زمان درخت ستونی j برگ $Y(j)$ام را انتخاب می‌کند.
- بنابراین $X(Y(j))$ انتخاب شده و به ریشه ارسال می‌شود.