۱. مقدمه‌ای بر مسیریابی و مدل‌سازی شبکه

استاد اشاره کردند که مسیریابی (Routing) فیلد بسیار گسترده‌ای است و میلیون‌ها الگوریتم وجود دارد که انتخاب آن‌ها به ساختار شبکه (وایرلس، سیمی، نوع روترها و...) بستگی دارد. اما به طور کلی الگوریتم‌ها به دو دسته تقسیم می‌شوند:

1. **Link State (LS):** حالت لینک.
2. **Distance Vector (DV):** بردار فاصله.

مدل‌سازی با گراف (Graph Theory)

برای درک بهتر رفتار شبکه و انجام محاسبات روی کاغذ، شبکه را به صورت **گراف** مدل‌سازی می‌کنند:

- **گراف (G):** مجموعه‌ای از نودها (V) و یال‌ها (E) است: $G = (V, E)$.
- **نودها (Nodes):** نشان‌دهنده روترها، سوئیچ‌ها یا سیستم‌ها هستند.
- **یال‌ها (Edges/Links):** نشان‌دهنده لینک‌های ارتباطی (کابل‌ها و کانال‌ها) هستند.
- **وزن (Weight):** عددی که روی یال‌ها قرار می‌گیرد. این عدد می‌تواند نشان‌دهنده هزینه، زمان، پهنای باند یا هر پارامتر اولویت‌بندی دیگری باشد. (در امتحان این اعداد داده می‌شوند).

---

۲. الگوریتم دایجسترا (Dijkstra)

این الگوریتم زیرمجموعه روش **Link State** است و هدف آن پیدا کردن **کوتاه‌ترین مسیر** (کم‌هزینه‌ترین) بین مبدأ و مقصد است.

- **هدف:** یافتن مسیر بهینه به طوری که تمام نودها دیده (Seen) شوند.
- **خروجی:** الگوریتم دایجسترا در نهایت یک **گراف کم‌هزینه (Least Cost Path Tree)** به ما می‌دهد، نه صرفاً یک خط مسیر. این گراف مشخص می‌کند که برای رسیدن به هر مقصدی، بهترین مسیر چیست.

روش حل (برای امتحان)

برای حل مسائل دایجسترا باید دو کار انجام دهید:

1. **رسم جدول:** جدولی که مراحل انتخاب مسیر و هزینه‌ها در آن نوشته شود.
2. **رسم گراف نهایی:** باید روی شکل اصلی مسیرهای انتخاب شده را پررنگ کنید یا گراف نهایی را جداگانه بکشید.

مراحل گام‌به‌گام:

1. از نود مبدأ شروع می‌کنیم.
2. هزینه رسیدن به **همسایه‌های مستقیم** را می‌نویسیم.
3. نودی که **کمترین هزینه** را دارد انتخاب می‌کنیم.
4. سپس از نود انتخاب شده، هزینه رسیدن به همسایه‌های جدید را محاسبه می‌کنیم. (هزینه کل = هزینه تا نود فعلی + هزینه یال جدید).
5. اگر مسیری با هزینه کمتر پیدا شد، آن را جایگزین می‌کنیم؛ در غیر این صورت مسیر قبلی می‌ماند.
6. این کار تا زمانی که تمام نودها به شبکه اضافه شوند ادامه می‌یابد.

> ***نکته امتحانی:*** *صرفاً نوشتن جواب آخر (مثلاً مسیر u-x-y-z) کافی نیست و نمره ندارد. باید جدول محاسبات و گراف نهایی وجود داشته باشد.*

۳. الگوریتم بلمن-فورد (Bellman-Ford)

این الگوریتم زیرمجموعه روش Distance Vector است و بر پایه یک فرمول ریاضی (معادله بلمن) کار می‌کند.

- ویژگی: برخلاف دایجسترا که گرافیکی‌تر است، این روش فرمول‌محور و بازگشتی (تو در تو) است.
- چالش: محاسبات آن سخت نیست (جمع و تفریق ساده)، اما چون فرمول‌ها تو در تو هستند، احتمال خطای محاسباتی یا نوشتاری زیاد است.

فرمول بلمن-فورد

فرمول کلی برای پیدا کردن کمترین هزینه از نود x به مقصد y به صورت زیر است:

$$d_x(y) = min_v\{c(x,v) + d_v(y)\}$$

- $d_x(y)$: فاصله (هزینه) از x تا y.
- min_v: انتخاب کمترین مقدار بین تمام همسایه‌های v.
- $c(x,v)$: هزینه (Cost) مستقیم بین نود x و همسایه‌اش v.
- $d_v(y)$: فاصله (هزینه) از همسایه v تا مقصد نهایی y.

روش حل:
برای حل، باید فرمول را برای تک‌تک همسایه‌های مبدأ بنویسید. چون مقدار $d_v(y)$ (فاصله همسایه تا مقصد) معمولاً مستقیم نیست، باید دوباره همین فرمول را برای همسایه بازنویسی کنید تا به مقصد برسید. در نهایت اعداد را جایگذاری کرده و کمترین مقدار را انتخاب می‌کنید.

۵. نکات کلیدی امتحانی

- در دایجسترا، هدف فقط پیدا کردن مسیر نیست، بلکه ساختن درختی است که همه نودها در آن دیده شده باشند (Seen).
- در حل مسائل، حواستان باشد که اعداد و وزن‌ها فقط برای محاسبه هستند و نباید بی‌دلیل با هم جمع شوند؛ منطق الگوریتم مهم است.
- در بلمن-فورد، دقت کنید که محاسبات را تمیز بنویسید تا در فرمول‌های تو در تو گم نشوید.