

طراحی و تحلیل الگوریتم ها

مدرس: سعدون عزیزی

s.azizi@uok.ac.ir

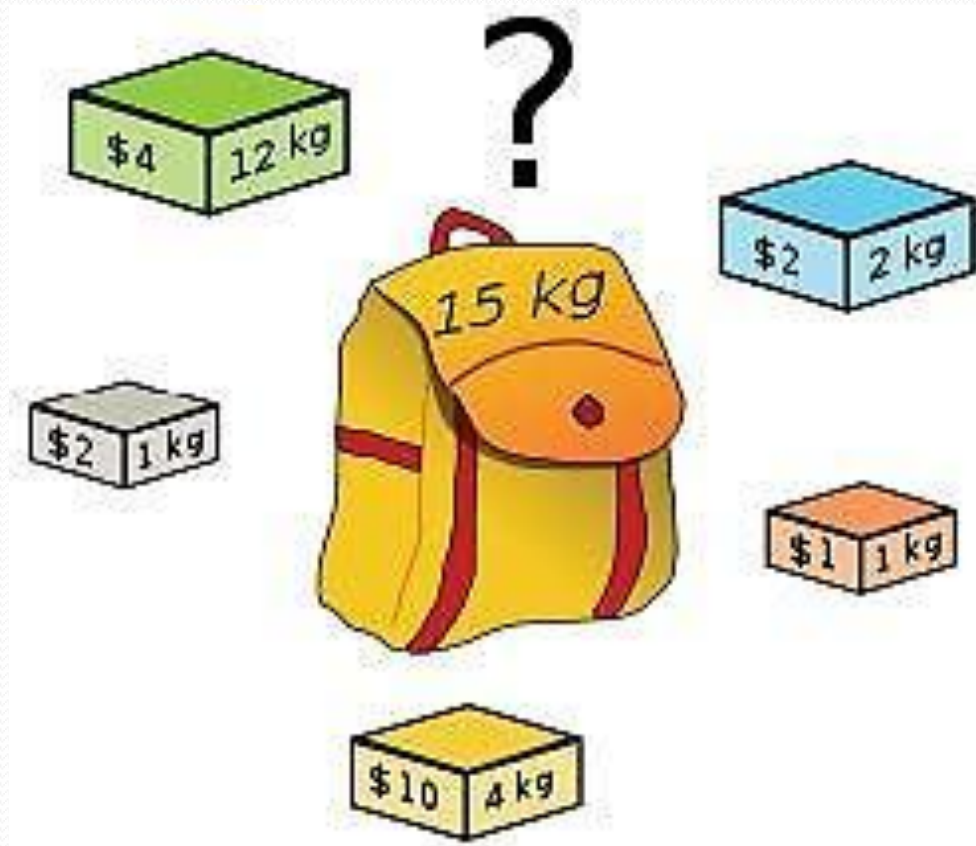
گروه مهندسی کامپیوتر

نیم سال دوم ۹۷-۹۶

تکنیک‌های طراحی الگوریتم‌ها

- تقسیم و حل
- برنامه‌ریزی پویا
- الگوریتم‌های حریصانه
- الگوریتم‌های عقبگرد
- الگوریتم‌های شاخه و حد

مسئله کوله‌پشتی (Knapsack Problem)



مسئله کوله‌پشتی

□ مسئله کوله‌پشتی ۰-۱ (0-1 knapsack problem): یک دزد که در حال دزدی از یک مغازه است، n فقره جنس پیدا می‌کند؛ قیمت جنس i ام v_i دلار و وزن آن w_i پوند است، که در آن v_i و w_i اعداد صحیح هستند. او می‌خواهد با ارزش‌ترین بار ممکن را بردارد، ولی کوله‌پشتی فقط به اندازه W پوند ظرفیت دارد که W یک عدد صحیح است. او باید کدام اجناس را بردارد؟ (از این رو به این مسئله، کوله‌پشتی ۰-۱ گفته می‌شود که در آن هر جنس یا برداشته می‌شود و یا خیر؛ به عبارت دیگر، دزد نمی‌تواند فقط قسمتی از یک جنس را بردارد. اجناس را به صورت شمش‌های طلا فرض کنید)

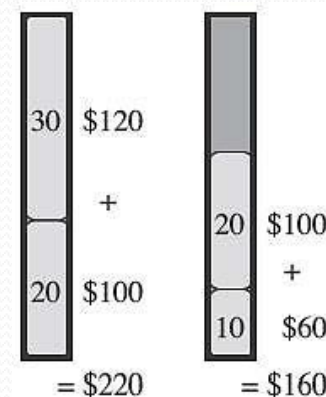
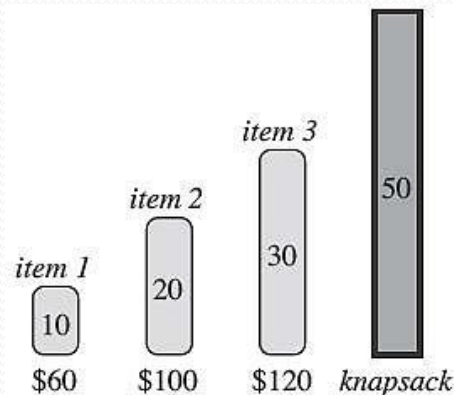
□ مسئله کوله‌پشتی کسری (fractional knapsack problem): صورت مسئله همانند مسئله فوق است، با این تفاوت که دزد می‌تواند کسری از اجناس را بردارد و مجبور نیست برای هر جنس یک انتخاب دودویی صفر و یک انجام دهد. (در این حالت، اجناس را به صورت گرد طلا فرض کنید)

مسئله کوله‌پشتی

❑ **سوال:** آیا می‌توان برای مسئله کوله پشتی ۰-۱ یک الگوریتم حریصانه ارائه داد؟ برای مسئله کوله پشتی کسری چطور؟

❑ یک استراتژی حریصانه واضح این است که قطعات با بیشترین ارزش زودتر از همه برداشته شوند. به همین منظور، ابتدا مقدار قیمت بر وزن، V_i/W_i ، را برای هر جنس محاسبه می‌کنیم.

❑ مثال نقض زیر نشان می‌دهد که این استراتژی حریصانه برای مسئله کوله‌پشتی ۰-۱ کار نمی‌کند.



مسئله کوله‌پشتی ۰-۱: برنامه‌ریزی پویا

□ فرض کنید که $P[i,j]$ نشان‌دهنده بیشترین سود بدست آمده از i قطعه اول باشد با این محدودیت که مجموع وزن آنها از j بیشتر نشود. بنابراین، هدف مسئله پیدا کردن $P[n,W]$ است. رابطه بازگشتی برای محاسبه $P[n,W]$ به صورت زیر است:

$$P[n, W] = \begin{cases} \max(P[n-1, W], P[n-1][W - w_n] + v_n) & \text{if } w_n \leq W \\ P[n-1][W] & \text{if } w_n > W \end{cases}$$

□ **تمرین:** نشان دهید که پیچیدگی این روش از مرتبه $O(\min(2^n, nW))$ است.

مسئله کوله‌پشتی ۰-۱: برنامه‌ریزی پویا

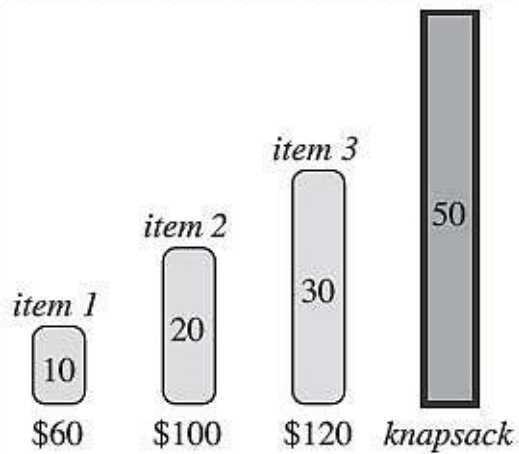
□ **مثال:** فرض کنید سه قطعه داریم، که اولی دارای وزن 5 پوند و ارزش \$50، دومی دارای وزن 10 پوند و ارزش \$60 و سومی با وزن 20 پوند با ارزش \$140 دلار است. ظرفیت کوله‌پشتی را نیز برابر $W=30$ در نظر بگیرید. با استفاده از رابطه بازگشتی ارائه شده در اسلاید قبل، راه‌حل بهینه را پیدا کنید.

□ **پاسخ:** \$200

مسئله کوله‌پشتی کسری: الگوریتم حریصانه

- الگوریتم حریصانه برای حل مسئله کوله‌پشتی کسری به صورت زیر است:
- ابتدا مقدار ارزش بر وزن، V_i/W_i ، را برای هر جنس محاسبه می‌کنیم.
- سپس، اجناس را به ترتیب نزولی مقدار ارزش بر وزن مرتب می‌کنیم.
- با دنبال کردن استراتژی حریصانه، در هر گام جنس با بیشترین نسبت ارزش بر وزن را انتخاب می‌کنیم.
- اگر باز هم جای خالی وجود نداشت، به مقدار ظرفیت باقی مانده از جنس بعدی (که هم‌اکنون بیشترین ارزش بر وزن را در بین بقیه اجناس دارد) برمی‌داریم.

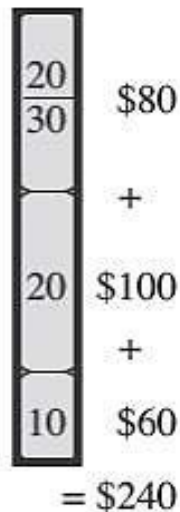
مسئله کوله‌پشتی کسری: مثال



$$\text{item1: } 60/10=6$$

$$\text{item2: } 100/20=5$$

$$\text{item3: } 120/30=4$$



مسئله کوله‌پشتی کسری: درستی الگوریتم

قضیه: الگوریتم حریصانه برای مسئله کوله پشتی کسری، همیشه حل بهینه را برمی‌گرداند.

اثبات به عنوان تمرین به دانشجو واگذار می‌گردد.