**پیاده‌سازی پروژه کوله پشتی 0-1 با دو روش برنامه نویسی پویا و راهبرد عقبگرد**

**نام و نام‌خانوادگی:**

**نام استاد:**

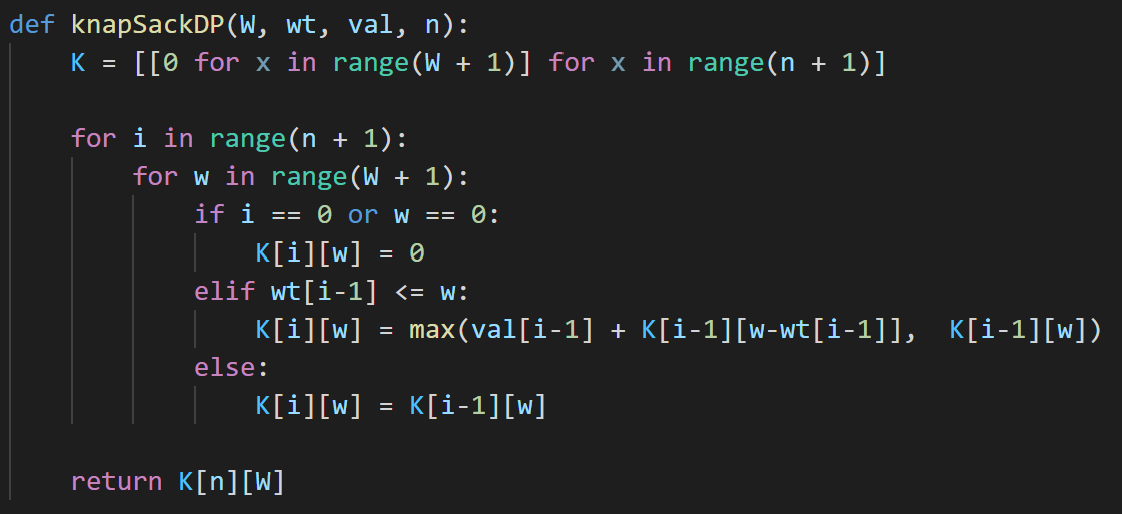
1. **شرح مسئله**

* هدف، قرار دادن این اشیا در کوله‌پشتی با ظرفیت W به صورتی است که مقدار ارزش بیشینه حاصل شود. به بیان دیگر، دو آرایه صحیح [val[0..n-1 و [wt[0..n-1 وجود دارند که به ترتیب مقادیر و وزن‌های تخصیص داده شده به n عنصر هستند.
* همچنین، یک عدد صحیح W نیز داده شده است که ظرفیت کوله پشتی را نشان می‌دهد. هدف، پیدا کردن زیرمجموعه‌ای با مقدار بیشینه []val است که در آن، مجموع وزن‌ها کوچک‌تر یا مساوی W باشد.
* امکان خورد کردن اشیا وجود ندارد و باید یک شی را به طور کامل انتخاب کرد و یا اصلا انتخاب نکرد. این گونه از مساله کوله پشتی را، «مساله کوله پشتی ۱-۰» می‌گویند.

[5.7 The 0-1 Knapsack Problem · Foundations of Algorithms Using C++ Pseudocod · 看云 (kancloud.cn)](https://www.kancloud.cn/leavor/cplusplus/630545)

1. **کد مربوط به هر دو روش (اسکرین از کدها)**

* **روش برنامه سازی پویا: (پایین به بالا)**



* **الگوریتم عقبگرد:**

در حل مسائل با رویکرد عقبگرد، تا زمانی که جستجوی تمام گره‌ها تمام نشود اطمینان نداریم که آیا گره‌ای در برگیرنده راه‌حل است یا خیر. پس اگر با رسیدن به گره‌ای سود بیشتری از سود فعلی به ما برسد، مقدار بیشترین سود را بروزرسانی می‌کنیم.

1. **تصویر خروجی برنامه در هردو روش (حداقل 5 ورودی مختلف، هرکدام شامل 5 کالا با ارزش و وزن مشخص)**
2. **مقایسه دو روش به صورت نمودار بر اساس زمان اجرا**
3. **پیچیدگی زمانی دو روش**
4. **تحلیل و مقایسه نتایج در چند سطر**
5. **نتیجه گیری (بر اساس تحلیل و مقایسه و پیچیدگی زمانی)، کدام روش بهتر است و چرا؟**