Algorytmy i struktury danych – wskazówki do realizacji zadań

Lista o zadanym rozmiarze

Listy w Pythonie są wewnętrznie reprezentowane w postaci tablic. Dodawanie elementów metodą `append` wiąże się z okresową realokacją pamięci i przeniesieniem wszystkich elementów. Jeżeli docelowy rozmiar listy jest znany, można ją wstępnie zaalokować, np.:

```
xs = [None] * 100
```

Korzystanie z wycinków list (ang. *slice*), wiąże się ze stworzeniem ich kopii (alokacją pamięci i skopiowaniem odpowiedniego fragmentu listy).

Maksymalna głębokość rekurencji

Interpreter Pythona ogranicza maksymalną liczbę wywołań funkcji na stosie, co chroni przed jego przepełnieniem w przypadku nieskończonej rekurencji. Wartość tego ograniczenia można uzyskać przy użyciu funkcji `sys.getrecursionlimit()`. Rezultatem jego przekroczenia będzie przerwanie działania programu.

W przypadku algorytmów rekurencyjnych, domyślne ograniczenie (zwykle 1000) może być zbyt małe. Można je zmienić przy użyciu funkcji `sys.setrecursionlimit(limit)`, gdzie `limit` będzie nową wartością ograniczenia.

Pomiar czasu wykonania

Do pomiaru czasu wykonania fragmentu kodu można użyć funkcji `time.process_time()` (dostępnej od wersji 3.3 Pythona), która zwraca czas procesora wykorzystany przez aktualny proces:

```
import time
start = time.process_time()  # pobierz aktualny czas
my_function()
stop = time.process_time()
print('Czas wykonania w sekundach:', stop - start)
```

Funkcja ta nie uwzględnia czasu, gdy proces został uśpiony (system operacyjny przydzielił czas innemu procesowi), dzięki czemu na pomiary nie będą miały wpływu inne uruchomione programy.

W takcie wykonywania mierzonego kodu może się uruchomić odśmiecanie pamięci (ang. *garbage collection*). Czas wykonania odśmiecania zostanie uwzględnione w pomiarze. Przed rozpoczęciem pomiaru należy wyłączyć odśmiecanie pamięci:

```
import gc
gc_old = gc.isenabled()  # pobierz aktualny stan odśmiecania
gc.disable()             # wyłącz odśmiecanie
# pomiar czasu wykonania...
if gc_old: gc.enable()   # przywróć pierwotny stan odśmiecania
```

Tworzenie wykresów

Do tworzenia wykresów można użyć biblioteki `Matplotlib` (https://matplotlib.org). Biblioteka ta oferuje dwa zestawy funkcji (API): *object-oriented* i *pyplot*, z których ten drugi daje mniejsze możliwości, ale jest prostszy w użyciu. Aby utworzyć plik PNG z wykresem, należy zamiast funkcji `show()` użyć `savefig('plik.png')`.

Generowanie wartości losowych

Do wygenerowania losowej liczby całkowitej można użyć funkcji `random.randint(a, b)`, gdzie `a` i `b` określają przedział (domknięty), z którego generowana będzie liczba. Np. wywołanie:

```
random.randint(0, 1000)
```

zwróci liczbę z zakresu od 0 do 1000.

Do wybrania losowej wartości z podanej listy można użyć funkcji `random.choice(lista)`. Np. wywołanie:

```
random.choice(['A', 'G', 'C', 'T'])
```

zwróci jeden z czterech napisów.