Problem kombinatoryczny może zostać opisany za pomocą skończonego zbioru parametrów wejściowych oraz warunków jakie mają spełniać dane wyjściowe. Problemy kombinatoryczne można podzielić na **problemy decyzyjne**, **optymalizacyjne** oraz **przeszukiwania**, przy czym podział ten nie jest rozłączny.

Problem decyzyjny definiuje się jako skończony zbiór parametrów (zbiory, grafy, funkcje, liczby etc.), które nie muszą mieć nadanych wartości oraz pytanie na które odpowiedź brzmi ''tak'' lub ''nie''. Przykładowo rozważmy problem trójpodziału z dziedziny problemów pakowania. Dla danego ograniczenia *B* i zbioru *A* składającego się z *3q* elementów o znanych wagach całkowitych, pytamy o istnienie podziału tego zbioru na *q* rozłącznych podzbiorów trójelementowych, takich że suma wag w każdym z nich jest równa *B*.

Problemy optymalizacyjne wymagają ekstremizacji określonej funkcji celu związanej z danym problemem kombinatorycznym. Natomiast **problem przeszukiwania** wymaga kompletnego rozwiązania lub odpowiedzi ''nie'' jeśli takowe nie istnieje.

Każdy problem decyzyjny i optymalizacyjny można przedstawić w postaci problemu przeszukiwania. Dla każdego problemu optymalizacyjnego a także dla każdego problemu przeszukiwania istnieje odpowiadający mu problem decyzyjny. Wersja decyzyjna danego problemu jest nie trudniejsza niż odpowiadająca jej pierwotna wersja optymalizacyjna lub przeszukiwania. Oznacza to, że jeśli w ''prosty sposób'' można rozwiązać problem przeszukiwania (lub optymalizacyjny) to można także ''prosto'' rozwiązać związany z nim problem decyzyjny. Jeśli natomiast problem decyzyjny jest obliczeniowo ''trudny'' to również odpowiadający mu problem przeszukiwania (lub optymalizacyjny) jest ''trudny''. Te klasy problemów mogą zatem być analizowane w ten sam sposób pod kątem ich ''natury obliczeniowej''. Przy czym, aby wykazać ''trudność'' problemu najczęściej wystarczy ograniczyć się do jego sformułowania decyzyjnego. Dziedzina nauki zajmująca się określaniem zasobów potrzebnych do rozwiązania różnych problemów obliczeniowych nosi miano teorii złożoności obliczeniowej.

Przykład problemu: ***Problem sortowania***

Dla danego ciągu liczb naturalnych (*I*) znaleźć ciąg uporządkowany rosnąco ($z \in Z_{\pi}(I)$).

Problemy kombinatoryczne (decyzyjne, optymalizacyjne oraz przeszukiwania) rozwiązywane są za pomocą pewnych procedur. Procedury te zwane są algorytmami. **Algorytm** to inaczej skończony ciąg jasno zdefiniowanych czynności, koniecznych do wykonania pewnego rodzaju zadań. Ma on przeprowadzić system z pewnego stanu początkowego do pożądanego stanu końcowego i może zostać zaimplementowany w postaci programu komputerowego lub innego urządzenia. Przykładem algorytmu, który rozwiązuje problem kombinatoryczny decyzyjny byłby program, który dla ciągu danych wejściowych zwraca odpowiedź „tak" lub „nie". Ważną cechą algorytmów jest ich **efektywność**, która często sprowadzana jest do ich złożoności czasowej (czasu potrzebnego na rozwiązanie problemu). Zbyt długi czas, który jest potrzebny algorytmowi do rozwiązania zadanego problemu może spowodować brak praktycznej przydatności tego algorytmu.

Funkcja złożoności obliczeniowej przyporządkowuje maksymalną liczbę kroków potrzebnych algorytmowi do rozwiązania zadanego problemu w zależności od rozmiaru instancji problemu (ilości danych wejściowych) oraz czasem również od ich wartości. Czas pracy danego algorytmu jest proporcjonalny do rozmiaru instancji rozwiązywanego problemu. Jeśli funkcji czasu nie można ograniczyć od góry wielomianem zależnym od rozmiaru instancji problemu, dany algorytm określa się jako **wykładniczy**. **Algorytmem wielomianowym** natomiast nazywamy algorytm, którego funkcja złożoności obliczeniowej jest $O(p(k))$, gdzie p jest pewnym wielomianem, a k rozmiarem rozwiązywanego konkretnego problemu. **Algorytm pseudowielomianowy** to taki algorytm, którego czas pracy można ograniczyć za pomocą wielomianu zależnego jednocześnie od rozmiaru konkretnego problemu oraz największej liczby występującej w jego instancji. Użyta powyżej notacja zwana **notacją dużego O** wyraża ogólną złożoność czasową zadanego algorytmu w przypadku pesymistycznym (czyli w przypadku konieczności analizy danych, które wymagają najdłuższego możliwego czasu obliczeń).

Problemy decyzyjne klasyfikuje się pod kątem ich złożoności obliczeniowej w podany poniżej sposób. **Klasa złożoności obliczeniowej P** oznacza, że problem ten może zostać rozwiązany w co najwyżej wielomianowym czasie przy użyciu deterministycznej maszyny Turinga (czyli dla określonego typu danych wejściowych, w określonym stanie przewidziana jest konkretna operacja). **Klasa złożoności NP** oznacza, że problem ten może zostać rozwiązany w czasie wielomianowym przez niedeterministyczną maszynę Turinga (dla określonych danych, w określonym stanie maszyna może postąpić na kilka różnych sposobów). Z definicji wynika, że każdy problem P należy również do klasy problemów NP.

Do klasy NP należą zarówno problemy łatwe (dla których istnieją algorytmy wielomianowe), jak i trudne (dla których takie algorytmy nie mogą zostać wprowadzone). **Klasa problemów NP-zupełnych** to klasa, do której należą najtrudniejsze problemy z klasy NP.