

# Instrukcja – Algorytmy i struktury danych

---

## Drzewa BST – przykładowe zastosowanie

---

Przy użyciu wcześniej zaimplementowanej struktury drzewa BST.

Zaimplementuj system zarządzający ruchem lotniczym na bardzo obciążonym lotnisku.

- System powinien pozwalać na „rezerwacje” planowanych lądowań,
- Jak tylko samolot ląduje – jego lądowanie jest usuwane z listy oczekiwanych zdarzeń,
- Rezerwacja lądowania odbywa się poprzez podanie przez pilota planowanego czasu lądowania. Jeśli w podanym czasie  $\pm 3$  min nie ma planowanych lądowań – dodać nowe zdarzenie, w przeciwnym wypadku – poinformować o niemożliwości,
- Możliwość wypisania całej listy planowanych lądowań na ekran.

Przykład:

- Planowane zdarzenia: (41, 47, 50, 58),
- Czas aktualny 37,
- Próba rezerwacji 48 – odmowa,
- Próba rezerwacji 20 – odmowa,
- Próba rezerwacji 54 – sukces.

Uwagi co do projektu:

- Dlaczego nie lista – sprawdzenie 3-minutowego odstępu wymaga ciągłego wyszukiwania elementów, w liście jest bardzo kosztowną operacją,
- Dlaczego nie tablica – konieczna jest struktura dynamiczna ze względu na ciągłą zmianę rozmiaru listy zdarzeń,
- Przy odpaleniu projektu jest pobierany aktualny czas z systemu (biblioteka `<ctime>`) <https://www.geeksforgeeks.org/print-system-time-c-3-different-ways/>,
- Dla uproszczenia ignorujemy wszystkie kwestie związane z aktualizacją czasu rzeczywistego na bieżąco. Czas jest aktualizowany tylko po obsłużeniu próby rezerwacji (po zakończeniu obsługi żądania rezerwacji pobieramy czas aktualny z systemu i usuwamy z listy zdarzeń wszystkie loty które już wylądowali).