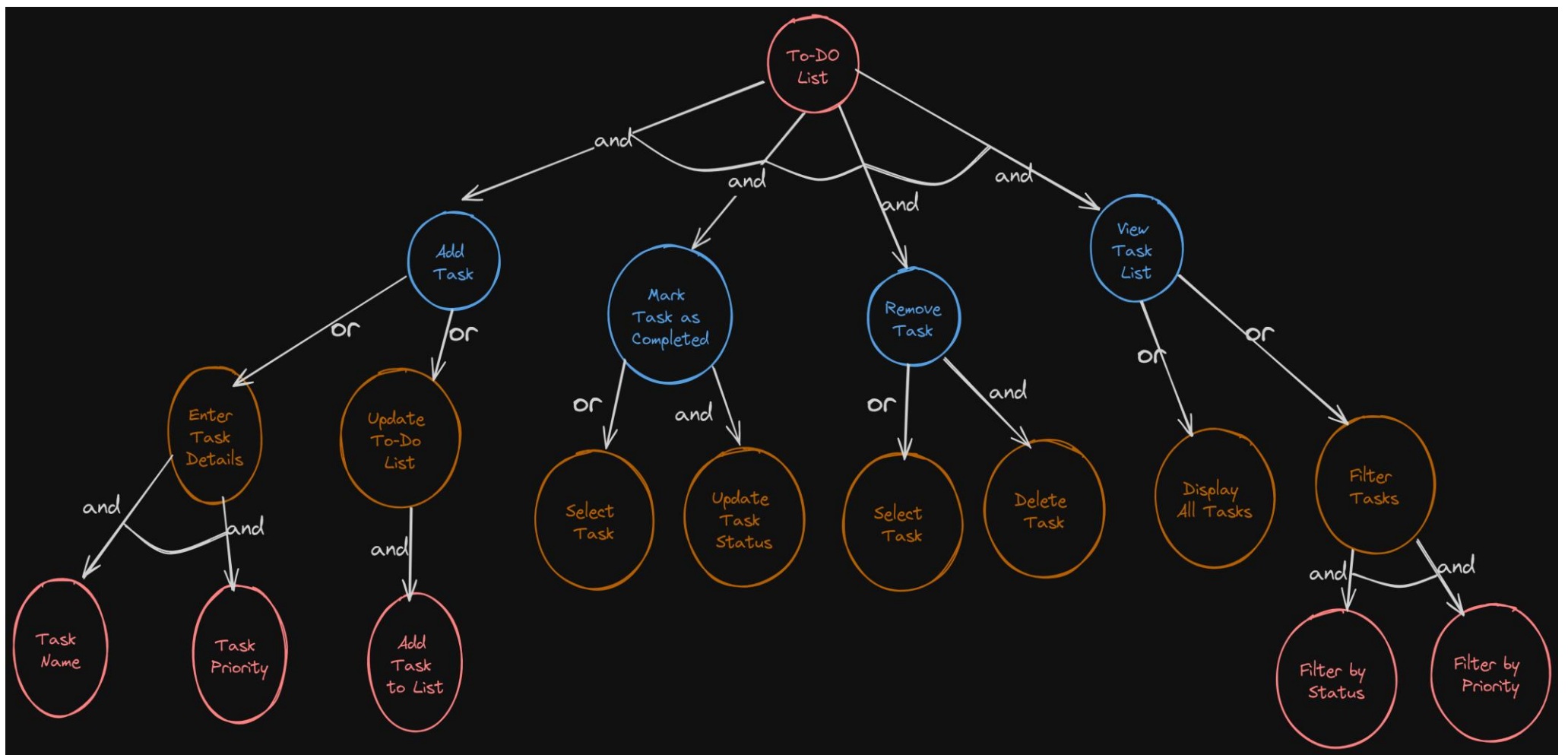


روال حل یک مسئله کامپیوتری با استفاده از درخت And-Or



ارائه یک مثال برای حل مسئله به روش IDEA

1. **Identify**: مسئله این که تعیین کنیم یا عدد ورودی ما عدد اول است یا خیر. عدد اول عددی است تنها بر خودش و عدد 1 بخش پذیر است.
2. **Develop**: اگر فرض کنیم عدد داده شده n باشد. یک راه حل این است که از عدد 2 تا $n - 1$ تمامی اعداد را یکی یکی با n تقسیم کنیم تا ببینیم که آیا بر عددی بخش پذیر است یا خیر. اگر حداقل بر یکی از اعداد بخش پذیر بود که عدد ورودی ما عدد اول نیست در غیر این صورت عدد اول می باشد.
3. **Evaluate**: حالا روش بالا را ارزیابی میکنیم. روش بالا بنظر یک روش منطقی و کاربردی می باشد و ظاهراً کار میکند. و دارای پیچیدگی زمانی $O(\sqrt{n})$ می باشد که برای بیشتر کاربردهای عملی به اندازه کافی بهینه است.
4. **Assess**: ما عواملی از جمله صحت، کارآمدی را در نظر گرفتیم. و بنظر می رسد که روش بالا درست است و برای تعیین اول بودن یک عدد بهینه است. اما با این این می تواند برای عددهای بسیار بزرگ مطلوب نباشد. برای رفع مشکل **performance** اعداد خیلی بزرگ پیشنهاد می شود که از الگوریتم **Sieve of Eratosthenes** استفاده شود.

روش سوم

1. **Problem**: کدام یک از رشته ها **Palindrome** هستند.

2. **Idea**: میتوانیم عدد را از ورودی بگیریم و آن را برعکس کنیم و در یک متغیر ثانویه ذخیره کنیم سپس متغیر ثانویه که رشته در آن برعکس شده است را با رشته اصلی مقایسه کنیم. اگر رشته ها با یکدیگر برابر بودند که ینی رشته مورد نظر ما یک **Palindrome** است. و اگر برابر نبودند یعنی نیست.

```
function isPalindrome(str) {
  let rev = str.split("").reverse().join("");

  if (rev === str) return true;
  else return false;
}
```

3.

4. **Debug**: تابع بالا درست است و کار میکند اما اگر رشته پاس داده شده شامل حروف بزرگ باشد ما به مشکل میخوریم پس باید تغییراتی بدهیم. پس تمام رشته مورد نظر را تبدیل به حروف کوچک میکنیم و سپس با رشته برعکس شده مقایسه میکنیم.

```
function isPalindrome(str) {
  let rev = str.split("").reverse().join("").toLowerCase();

  if (rev === str.toLowerCase()) return true;
  else return false;
}
```

5. **Assess**: بعد از تست تابع مورد نظر با ورودی های زیر و دریافت پاسخ صحیح. به درستی برنامه خود پی میبریم.

```
test("Is Palindrome", () => {
  expect(isPalindrome("MoM")).toBe(true);
  expect(isPalindrome("tenet")).toBe(true);
  expect(isPalindrome("anna")).toBe(true);
  expect(isPalindrome("ali")).toBe(false);
  expect(isPalindrome("TeNeT")).toBe(true);
  expect(isPalindrome("aNNa")).toBe(true);
  expect(isPalindrome("WoW")).toBe(true);
  expect(isPalindrome("ReFEr")).toBe(true);
  expect(isPalindrome("737")).toBe(true);
  expect(isPalindrome("12344321")).toBe(true);
  expect(isPalindrome("Hello")).toBe(false);
  expect(isPalindrome(123456)).toBe(false);
  expect(isPalindrome(12321)).toBe(true);
});
```