تمرین شماره ۵ مالتی مدیا

الف)

سری فوریه و تبدیل فوریه هر دو ابزاری برای انتقال مجموعه‌ای داده‌ها از دامنه فضا/زمان به دامنه فرکانس و نمایش متفاوت آن‌ها هستند. ورودی سری فوریه همواره بایستی پریودیک (متناوب) باشد. فارغ از پیوسته یا گسسته بودن تابع ورودی، سری فوریه همواره نمایشی گسسته (ضرایب سری فوریه a_k) از تابع دریافتی ارائه می‌نماید. هنگامی که ورودی تابعی گسسته و پریودیک باشد، خروجی نیز گسسته و پریودیک خواهد بود. اما هنگامی که ورودی پیوسته و پریودیک باشد، خروجی گسسته و غیرپریودیک خواهد بود.

در طرف مقابل، تابع ورودی تبدیل فوریه الزامی به پریودیک بودن ندارد و می‌تواند غیرپریودیک باشد. اما همانند سری فوریه، بنا بر گسسته یا پیوسته بودن ورودی، پریودیک بودن خروجی متغییر خواهد بود. هنگامی که تابع ورودی، تابعی پیوسته و غیرپریودیک باشد، معادل آن در دامنه فرکانس تابعی نیز تابعی پیوسته و غیرپریودیک خواهد بود. اما اگر تابع ورودی تابعی گسسته و غیرپریودیک باشد، خروجی پیوسته و پریودیک خواهد بود.

رابطه‌های زیر مربوط به سری فوریه گسسته زمان است:

$$x[n] = \sum_{m=0}^{N-1} \tilde{X}[m] e^{j2\pi\frac{mn}{N}} \qquad \tilde{X}[m] = \frac{1}{N}\sum_{n=0}^{N-1} x[n] e^{-j2\pi\frac{mn}{N}}$$

رابطه‌های زیر مربوط به سری فوریه پیوسته زمان است:

$$x(t) = \sum_{m=-\infty}^{\infty} X[m] e^{j2\pi m\frac{t}{T}} \qquad X[m] = \frac{1}{T}\int_{-T/2}^{T/2} x(t) e^{-j2\pi m\frac{t}{T}} dt$$

رابطه‌های زیر مربوط به تبدیل فوریه گسسته زمان است:

$$x[n] = \int_{-1/2}^{1/2} \tilde{X}(\varphi) e^{j2\pi\varphi n} d\varphi \qquad \tilde{X}[\varphi] = \sum_{n=-\infty}^{\infty} x[n] e^{-j2\pi\varphi n}$$

و رابطه‌های زیر نمایشگر تبدیل فوریه پیوسته زمان هستند:

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi ft} df \qquad X(f) = \int_{-\infty}^{\infty} x(t) e^{j2\pi ft} dt$$

ب)

هدف اصلی به وجود آمدن الگوریتم *Fast Fourier Transform (FFT)* همانطور که نام آن مشخص است، تسریع فرآیند محاسبه تبدیل فوریه گسسته زمان بوده است.

برای محاسبه تبدیل فوریه گسسته در حالت عادی نیاز به محاسبه حاصل ضرب ماتریس تبدیل در تابع ورودی می‌باشد. ماتریس تبدیل برای ورودی با n داده به صورت زیر خواهد بود:

$$F_n = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w & w^2 & & w^{n-1} \\ 1 & w^2 & w^4 & & w^{2(n-1)} \\ \vdots & & & \ddots & \vdots \\ 1 & w^{n-1} & w^{2(n-1)} & \cdots & w^{(n-1)^2} \end{bmatrix}$$

و برای محاسبه این حاصل نیاز به انجام n بار عملیات شامل n مرتبه ضرب خواهیم بود. در نتیجه پیچیدگی زمانی این روش برای محاسبه تبدیل فوریه برابر با $O(n^2)$ خواهد بود که مقدار قابل توجهی است.

برای کاهش این زمان الگوریتم FFT به وجود آمده است. ایده این الگوریتم تجزیه ماتریس تبدیل و شکستن آن به دو ماتریس کوچکتر با خاصیت راحتی ضرب و همچنین تغییر ترتیب داده‌های ورودی می‌باشد.

این الگوریتم داده‌های ورودی را بر اساس اندیس آن‌ها به دو دسته زوج و فرد تقسیم می‌نماید. و از طرفی ماتریس تبدیل فوق را به دو ماتریس زیر تبدیل می‌نماید:

$$F_{2n} = \begin{bmatrix} I & D \\ I & -D \end{bmatrix} \begin{bmatrix} F_n & 0 \\ 0 & F_n \end{bmatrix}$$

که در آن I نشان دهنده ماتریس همانی از سایز n می باشد.

و ماتریس D نشان دهنده ماتریس قطری زیر خواهد بود:

$$D = \begin{bmatrix} 1 & & & & \\ & w & & & \\ & & w^2 & & \\ & & & \ddots & \\ & & & & w^{n-1} \end{bmatrix}$$

با توجه به قطری بودن این دو ماتریس هزینه ضرب‌ها از مرتبه $O(n)$ خواهد بود. اما می‌توان این فرآیند را به صورت بازگشتی برای هر کدام از ماتریس‌های F_n نیز ادامه داد و هرکدام از آن‌ها را به ماتریس‌های قطری و دو $F_{n/2}$ تبدیل کرد و این فرآیند را برای $log_2(n)$ بار تکرار کرد تا به ماتریس‌های مربعی با سایز ۲ برسیم. همچنین باید علاوه بر تجزیه این ماتریس‌ها داده‌های ورودی را به تعداد $log_2(n)$ بار بر اساس اندیس آن‌ها تقسیم‌بندی کرد. در نهایت هزینه زمانی این فرآیند $O(n.log_2(n))$ خواهد بود.

در نتیجه در الگوریتم FFT با استفاده از تجزیه متوالی ماتریس تبدیل و تغییر ترتیب داده‌های ورودی توانستیم از مرتبه $O(n^2)$ به مقدار $O(n.log_2(n))$ برسیم که خصوصا برای n های بسیار بزرگ بهبود زمانی بسیار مناسبی خواهد بود.